

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Т. Ф. ГОРБАЧЕВА»
Филиал КузГТУ в г. Белово

Кафедра информационных технологий и гуманитарных дисциплин

**МДК 02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Методические указания к лабораторно-практическим работам
для обучающихся очной формы обучения

09.02.07 «Информационные системы и программирование»

Составитель Белугина С.В.

Рассмотрены и утверждены
на заседании кафедры
Протокол № 5 от 08.12.2022г.
Рекомендованы для использования в
образовательном процессе
учебно-методической комиссией по
специальности 09.02.07 «Информационные
системы и программирование»
Протокол № 4 от 09.12.2022г.

Составил: ст. преподаватель Белугина С.В.

Обсуждено на заседании кафедры информационных технологий и гуманитарных дисциплин

Зав. кафедрой Верчагина И.Ю

Согласовано учебно-методической комиссией по специальности 09.02.07
«Информационные системы и программирование»

Председатель учебно-методической комиссии Колечкина И.П

СОДЕРЖАНИЕ

Введение	3
Общие методические рекомендации и указания по выполнению практических занятий	4
Лабораторная работа 1 Разработка структуры проекта	6
Лабораторная работа 2 Разработка модульной структуры проекта (диаграммы модулей)	12
Лабораторная работа 3 Разработка перечня артефактов и протоколов проекта	14
Лабораторная работа 4 Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)	20
Лабораторная работа 5 Разработка и интеграция модулей проекта (командная работа)	25
Лабораторная работа 6 Отладка отдельных модулей программного проекта	27
Лабораторная работа 7 Организация обработки исключений	33
Практическое занятие 1 Применение отладочных классов в проекте	35
Лабораторная работа 8 Отладка проекта	37
Практическое занятие 2 Инспекция кода модулей проекта	44
Лабораторная работа 9 Тестирование интерфейса пользователя средствами инструментальной среды разработки	48
Лабораторная работа 10 Разработка тестовых модулей проекта для тестирования отдельных модулей	49
Лабораторная работа 11 Выполнение функционального тестирования	52
Лабораторная работа 12 Тестирование интеграции	55
Практическое занятие 3 Документирование результатов тестирования	56
Список используемых источников	58

ВВЕДЕНИЕ

Профессиональный модуль ПМ.02 «Осуществление интеграции программных модулей» является частью профессионального цикла основной образовательной программы в соответствии с ФГОС по специальности 09.02.07 «Информационные системы и программирование». ПМ.02 «Осуществление интеграции программных модулей» обеспечивает формирование профессиональных и общих компетенций по всем видам деятельности ФГОС по специальности 09.02.07 «Информационные системы и программирование».

Целью освоения междисциплинарного курса МДК 02.02 «Инструментальные средства разработки программного обеспечения» является приобретение обучающимися знаний в области формирования требований к ИС и тестирования программного обеспечения (ПО).

Основными задачами изучения МДК 02.02 «Инструментальные средства разработки программного обеспечения», являются:

1. Изучение основных принципов процесса разработки программного обеспечения.
2. Изучение инструментальных средств отладки программных продуктов.
3. Использовать приемы работы в системах контроля версий.
4. Умение выполнять отладку, используя методы и инструменты среды программирования.
5. Умение выполнять ручное и автоматизированное тестирование программного модуля.
6. Умение выполнять тестирование интеграции.
7. Умение анализировать проектную и техническую документацию.

МДК 02.02 содержит два раздела. Сборник состоит из введения, описания трех практических занятий и двенадцати лабораторных работ, списка используемой литературы.

Общее количество часов на практические работы – 36.

Методические указания к лабораторно-практическим работам соответствуют разделам и темам рабочей программы, содержат задания, технологию выполнения работы, вопросы для самоконтроля изученного материала.

Обучающиеся имеют возможность видеть конечную цель своего труда, планировать уровень его сложности и прогнозировать результат.

Практические работы проводятся в порядке изучения программного материала. Для более эффективного выполнения практических работ необходимо заранее повторить соответствующий теоретический материал по учебным пособиям, на занятии тщательно ознакомиться с содержанием работы и программным обеспечением компьютера.

ОБЩИЕ МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ И УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ

1.1 Подготовка к лабораторно-практическим работам

1. Внимательно ознакомиться с описанием соответствующей практической или лабораторной работы и установить, в чем состоит ее основная цель и задача;
2. По лекционному курсу и соответствующим литературным источникам изучить теоретическую часть, относящуюся к данному практическому занятию.

1.2 Выполнение лабораторно-практических работ

Успешное выполнение работы может быть достигнуто в том случае, если обучаемый представляет себе цель выполнения лабораторной или практической работы, поэтому важным условием является тщательная подготовка к лабораторно-практической работе.

1.3 Как пользоваться методическими указаниями

- внимательно прочтите текст задания, который часто представляет собой алгоритм будущих технологических действий; в каждом новом задании для наглядности описана технология выполнения для достижения поставленной в задании цели.
- по окончании выполнения предложенных заданий необходимо ответить на вопросы самоконтроля в конце каждой практической работы.

1.4 Оформление отчетов по лабораторно-практическим работам

Требования к структуре и оформлению отчета по лабораторно-практическим работам:

- отчет по лабораторно-практическим работам состоит из титульного листа, отчетов по выполненным работам
- каждая работа содержит цель, ход работы, ответы на вопросы самоконтроля и вывод о проделанной работе;
- отчет выполняют, руководствуясь следующими положениями:
 - отчет выполнять на листах формата А4;
 - записать на первом листе цель и постановку задачи;
 - оформлять работу шрифтом Times New Roman.

Форма отчетности – письменный отчет о лабораторной или практической работе на электронном носителе с устной защитой преподавателю.

ПЕРЕЧЕНЬ ЛАБОРАТОРНО- ПРАКТИЧЕСКИХ РАБОТ

№ темы	№ работы	Темы	Кол-во часов
Тема 1	1	Лабораторная работа 1 Разработка структуры проекта	4
	2	Лабораторная работа 2 Разработка модульной структуры проекта (диаграммы модулей)	2
	3	Лабораторная работа 3 Разработка перечня артефактов и протоколов проекта	2
	4	Лабораторная работа 4 Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)	2
	5	Лабораторная работа 5 Разработка и интеграция модулей проекта (командная работа)	2
	6	Лабораторная работа 6 Отладка отдельных модулей программного проекта	4
	7	Лабораторная работа 7 Организация обработки исключений	2
Тема 2	1	Практическое занятие 1 Применение отладочных классов в проекте	2
	8	Лабораторная работа 8 Отладка проекта	4
	2	Практическое занятие 2 Инспекция кода модулей проекта	2
	9	Лабораторная работа 9 Тестирование интерфейса пользователя средствами инструментальной среды разработки	2
	10	Лабораторная работа 10 Разработка тестовых модулей проекта для тестирования отдельных модулей	2
	11	Лабораторная работа 11 Выполнение функционального тестирования	2
	12	Лабораторная работа 12 Тестирование интеграции	2
		Практическое занятие 3 Документирование результатов тестирования	2
Итого			36

ЛАБОРАТОРНАЯ РАБОТА №1 (4 часа)

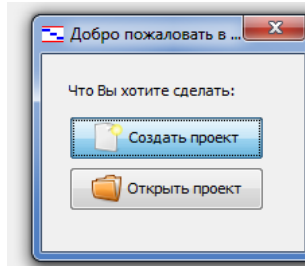
Тема: Разработка структуры проекта

Цель занятия: создание структуры проекта и заполнение базовой информации о проекте.

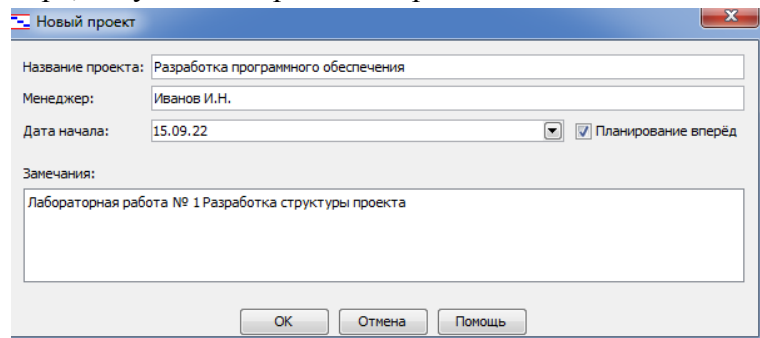
Технология работы:

Задание 1

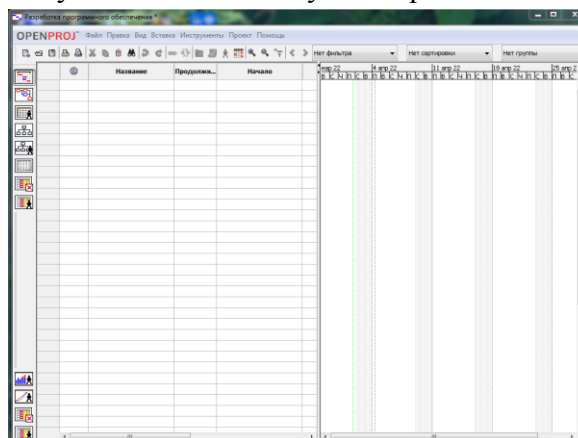
1. Запустите программу Open Project.
2. Изучите интерфейс программы.
3. Создайте новый проект.



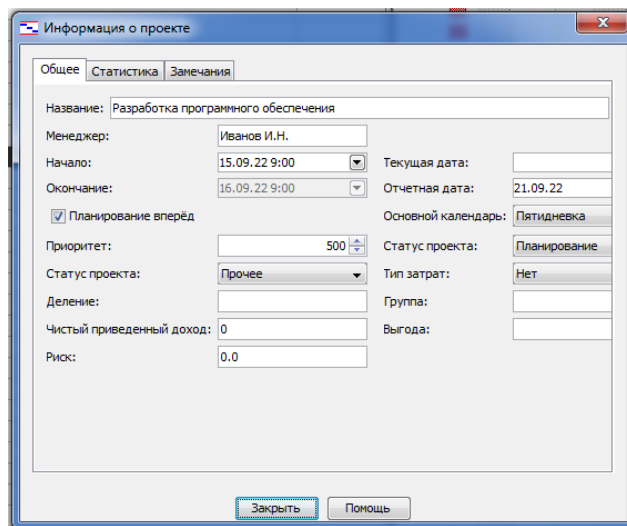
4. В появившемся окне следует заполнить данные о новом проекте. Необходимо указать название проекта, автора, дату начала проекта и краткие сведения.



5. После нажатия на кнопку «Ок» появится пустой проект.



6. После создания проекта необходимо настроить его основные параметры. Для настройки планирования от начальной даты выберите в меню **Проект** пункт **Сведения о проекте**. В появившемся окне ставим *Дату начала*. Да окончания будет рассчитана далее автоматически.

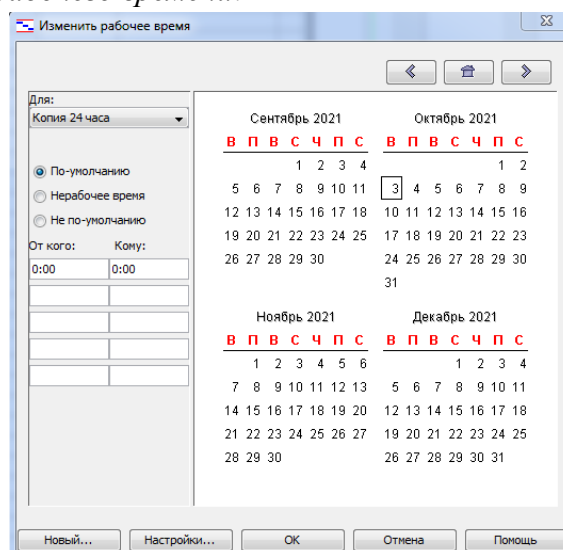


7. Выберите календарь для проекта.

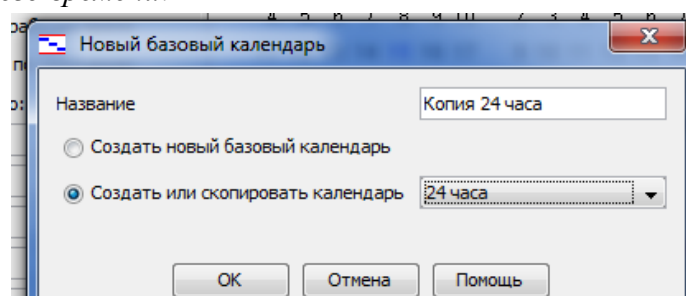
В состав пакета MS Project входит три базовых календаря – стандартный, ночная смена и 24 часа.

В *стандартном календаре* рабочий день начинается с 8:00 и заканчивается в 17:00 с обеденным перерывом с 12:00 до 13:00. Рабочая неделя начинается с понедельника и заканчивается в пятницу. Это календарь, применяемый по умолчанию. В *календаре ночной смены* рабочий день начинается с 23:00 и заканчивается в 8:00 с часовым перерывом с 03:00 до 04:00. В *календаре «24 часа»* рабочее время продолжается круглые сутки без выходных и обеденных перерывов.

Базовые календари можно редактировать для этого в меню *Инструменты* необходимо выбрать пункт *Изменение рабочего времени*.



Можно создать новое базовое расписание. Для этого в окне *Изменение рабочего времени* нажимаем кнопку *Создать*. В появившемся окне выбираем создание нового календаря на основе стандартного или создание копии любого другого календаря. Значения рабочего времени для вновь созданного календаря могут также быть отредактированы через окно *Изменение рабочего времени*.



8. Добавьте в план проекта следующие фазы жизненного цикла ПО.

- Анализ и требования к программному обеспечению
- **Проектирование**
- Разработка
- Тестирование
- Документация

	Название	Продолжение
1	Анализ и требования к программному продукту	
2	Проектирование	
3	Разработка	
4	Тестирование	
5	Документация	

9. Для каждой фазы укажите в плане проекта завершающие задачи (вехи проекта):

- Анализ завершен
- Проектирование завершено
- Разработка завершена
- Тестирование завершено
- Документация завершена
- Разработка программного обеспечения завершена

Для вставки пустых строк между названиями фаз проекта используйте клавишу Ins или команду Вставка / Новая задача. Для того, чтобы задача стала завершающей (вехой) следует установить ее длительность равной 0 дней.

	Название	Продолжительность	12 сен 22
1	Анализ и требования к программному продукту	1 день?	
2	Анализ завершен	0 дней?	◆ 15.09.22
3	Проектирование	1 день?	
4	Проектирование завершено	0 дней?	◆ 15.09.22
5	Разработка	1 день?	
6	Разработка завершена	0 дней?	◆ 15.09.22
7	Тестирование	1 день?	
8	Тестирование завершено	0 дней?	◆ 15.09.22
9	Документация	1 день?	
10	Разработка программного обеспечения завершена	0 дней?	◆ 15.09.22

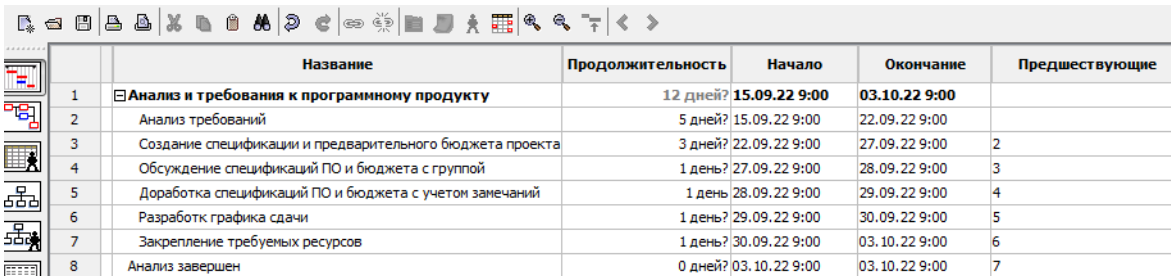
10. Декомпозируйте план проекта, добавив в него следующие задачи для фазы *Анализ и требования к программному обеспечению*:

- Анализ требований
- Создание спецификации и предварительного бюджета проекта
- Обсуждение спецификаций программного обеспечения и бюджета с группой
- Доработка спецификаций программного обеспечения и бюджета с учетом замечаний
- Разработка графика сдачи
- Закрепление требуемых ресурсов

11. Установите длительность задач фазы *Анализ и требования к программному обеспечению* в соответствие со следующей таблицей:

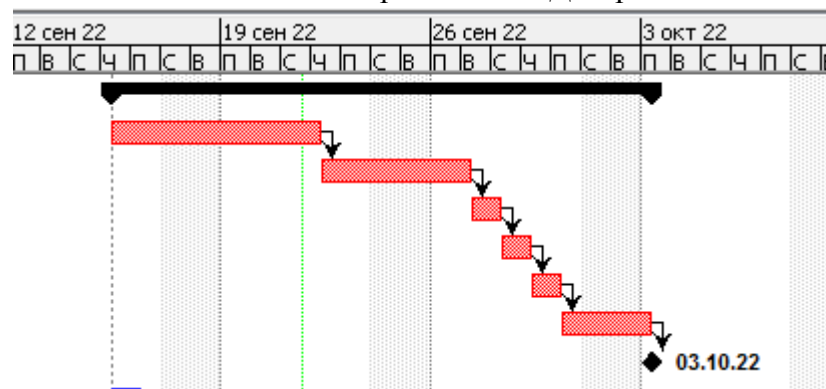
Задача	Длительность
Анализ требований	5 дней
Создание спецификации и предварительного бюджета проекта	3 дня
Обсуждение спецификаций программного обеспечения и бюджета с группой	4 часа
Доработка спецификаций программного обеспечения и бюджета с учетом замечаний	1 день
Разработка графика сдачи	1 день
Закрепление требуемых ресурсов	1 день

12. Установите связи между задачами.

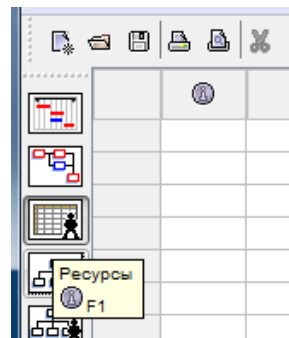


	Название	Продолжительность	Начало	Окончание	Предшествующие
1	Анализ и требования к программному продукту	12 дней?	15.09.22 9:00	03.10.22 9:00	
2	Анализ требований	5 дней?	15.09.22 9:00	22.09.22 9:00	
3	Создание спецификации и предварительного бюджета проекта	3 дней?	22.09.22 9:00	27.09.22 9:00	2
4	Обсуждение спецификаций ПО и бюджета с группой	1 день?	27.09.22 9:00	28.09.22 9:00	3
5	Доработка спецификаций ПО и бюджета с учетом замечаний	1 день?	28.09.22 9:00	29.09.22 9:00	4
6	Разработк графика сдачи	1 день?	29.09.22 9:00	30.09.22 9:00	5
7	Закрепление требуемых ресурсов	1 день?	30.09.22 9:00	03.10.22 9:00	6
8	Анализ завершен	0 дней?	03.10.22 9:00	03.10.22 9:00	7

13. Созданные задачи и их связи отобразились на Диаграмме Ганта.



14. Добавьте к задачам исполнителей. Исполнители являются ресурсами проекта. Перейдем к вкладке «Ресурсы», расположенной на боковой панели.



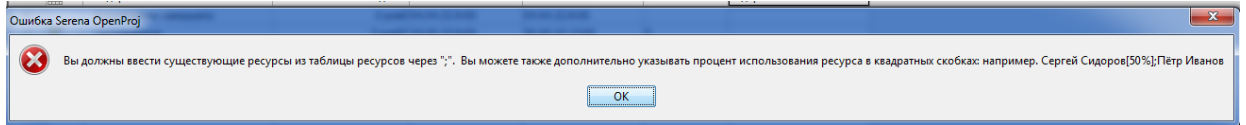
15. В диалоговом окне появится таблица с ресурсами проекта. Добавить в проект 10 исполнителей.

	И	Название	RBS	Тип	E-mail адрес	Ед.изм. материалов	Инициалы	Группировать
1		Иванов		Работа			И	Руководство
2		Петров		Работа			П	Руководитель проекта
3		Сидоров		Работа			С	Руководитель проекта
4		Жуков		Работа			Ж	Разработчик

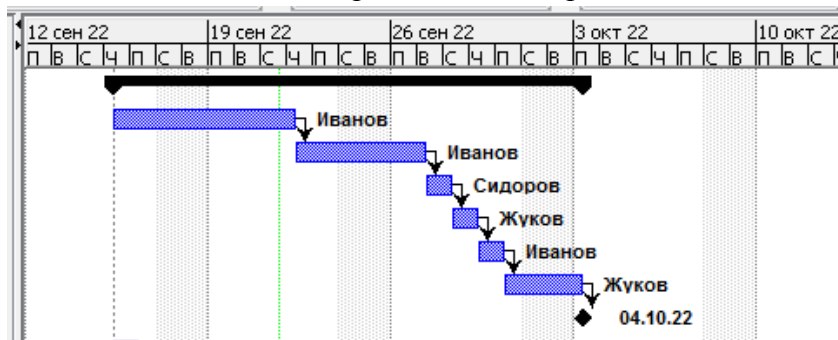
16. Назначить исполнителей проекта, вписав их имена в соответствующее поле.

	⊖	Название	Продолжительность	Начало	Окончание	Предшествующие	Название
1		☐ Анализ и требования к программному продукту	12 дней?	15.09.22 9:00	03.10.22 9:00		
2		Анализ требований	5 дней?	15.09.22 9:00	22.09.22 9:00		Иванов
3		Создание спецификации и предварительного бюджета проекта	3 дней?	22.09.22 9:00	27.09.22 9:00	2	Иванов
4		Обсуждение спецификаций ПО и бюджета с группой	1 день?	27.09.22 9:00	28.09.22 9:00	3	Сидоров
5		Доработка спецификаций ПО и бюджета с учетом замечаний	1 день?	28.09.22 9:00	29.09.22 9:00	4	Жуков
6		Разработк графика сдачи	1 день?	29.09.22 9:00	30.09.22 9:00	5	Иванов
7		Закрепление требуемых ресурсов	1 день?	30.09.22 9:00	03.10.22 9:00	6	Жуков
8		Анализ завершен	0 дней?	03.10.22 9:00	03.10.22 9:00	7	

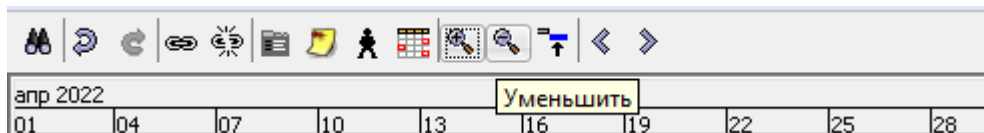
17. В случае если попытаться ввести имя несуществующего ресурса, программа выведет сообщение об ошибке.



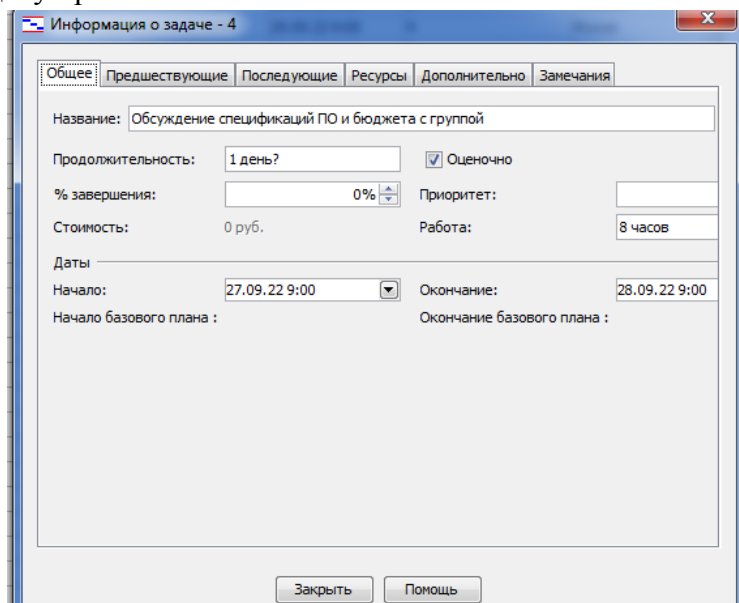
18. Назначенные исполнители отобразятся на Диаграмме Ганта.



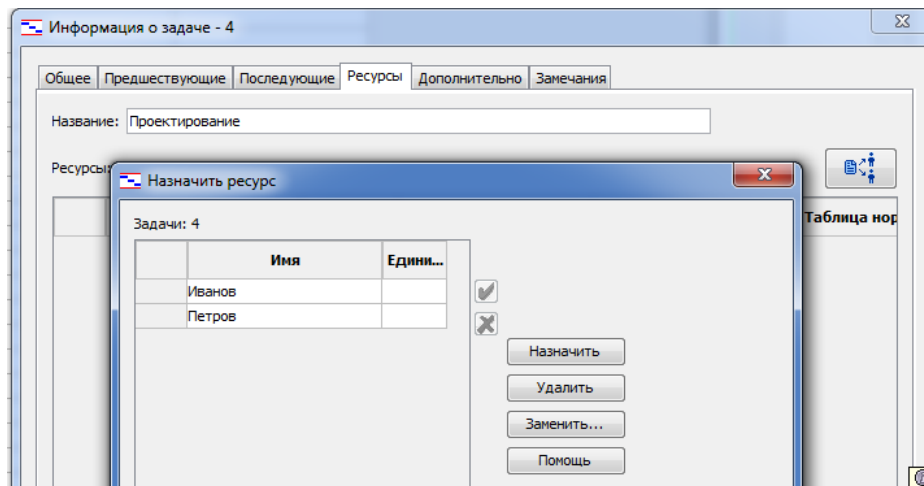
19. В программе присутствует возможность фильтрации задач на Диаграмме Ганта, а также изменение масштаба отображения временной шкалы.



20. Двойной клик мышью по задаче вызывает окно свойств этой задачи. Вкладка общие позволяет получить информацию о названии, продолжительности и проценте выполнения. Откройте любую задачу проекта.



21. Вкладки «Предыдущие» и «Последующие» позволяют получить информацию о связанных задачах. Интерес может представлять вкладка «Ресурсы», на которой будут отображаться исполнители задачи.



22. Аналогично декомпозируйте план проекта, добавив в него следующие задачи и подфазы:

- фаза Проектирование:
 - Разработка функциональных спецификаций
 - Разработка прототипа на основе функциональной спецификации
 - Ревизия функциональных спецификаций
 - Доработка функциональных спецификаций с учетом замечаний
- фаза Разработка:
 - Определение параметров модульной и уровневой архитектуры
 - Назначение персонала для разработки
 - Разработка кода
- фаза Тестирование:
 - подфаза Тестирование модулей:
 - Тестирование модулей компонента в соответствии со спецификацией продукта
 - Выявление недостатков в спецификациях продукта
 - Изменение кода
 - Повторное тестирование измененного кода
 - Тестирование модулей завершено (веха подфазы);
 - подфаза Тестирование интеграции:
 - Тестирование интеграции модулей
 - Выявление недостатков в спецификациях
 - Изменение кода
 - Повторное тестирование измененного кода
 - Тестирование интеграции завершено (веха подфазы);
- фаза Документация:
 - Разработка справки
 - Ревизия справки
 - Доработка справки с учетом замечаний
 - Разработка руководства пользователя
 - Ревизия всей документации для пользователей
 - Доработка документации для пользователей с учетом замечаний

23. Установите длительность задач плана проекта в соответствие со следующей таблицей:

Задача	Длительность
Разработка функциональных спецификаций	5 дней
Разработка прототипа на основе функциональной спецификации	4 дня
Ревизия функциональных спецификаций	2 дня
Доработка функциональных спецификаций с учетом замечаний	4 часа
Определение параметров модульной и уровневой архитектуры	4 часа
Назначение персонала для разработки	1 день
Разработка кода	15 дней
Тестирование модулей компонента в соответствии со спецификацией продукта	2 дня
Выявление недостатков в спецификациях продукта	3 дня
Изменение кода	3 дня
Повторное тестирование измененного кода	2 дня
Тестирование интеграции модулей	5 дней
Выявление недостатков в спецификациях	2 дня
Изменение кода	3 дня
Повторное тестирование измененного кода	2 дня
Разработка справки	3 недели
Ревизия справки	3 дня
Доработка справки с учетом замечаний	2 дня
Разработка руководства пользователя	3 недели
Ревизия всей документации для пользователей	2 дня
Доработка документации для пользователей с учетом замечаний	2 дня

24. Установите связи между задачами

25. Назначьте исполнителей проекта, вписав их имена в соответствующее поле.

26. Все изменения отобразятся на Диаграмме Ганта.

27. Создайте сетевой график с помощью пиктограммы



28. После завершения планирования проекта сохраните результаты работы.

Контрольные вопросы:

1. Что такое Open Project?
2. Что представляет собой проект в Open Project?
3. Что представляет собой работа в Open Project?
4. Что обычно подразумеваются под ресурсами?
5. Какие существуют базовые календари в программе MS Project?
6. Что такое Диаграмма Ганта?
7. Какие этапы присутствуют в календарном плане проекта?

ЛАБОРАТОРНАЯ РАБОТА № 2 (2 часа)

Тема: Разработка модульной структуры проекта (диаграммы модулей)

Цель: Изучить процесс разработки модульной структуры программного обеспечения, осуществляемого с помощью структурных карт Константайна.

Задания - Разработать модульную структуру программного модуля.

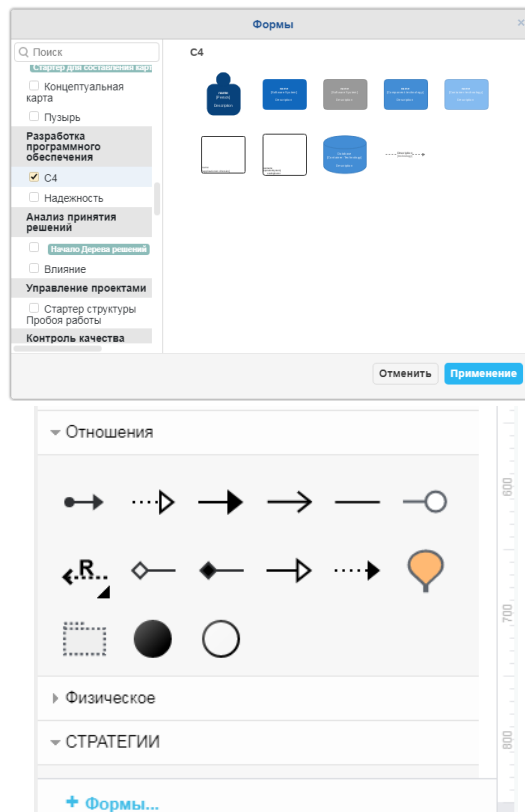
Технология выполнения:

- 1 Запустите программу Visual Paradigm Online

(<https://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard>)

2 Выберите шаблоны группы Диаграммы: SDL Diagram, ArchiMate Tutorial Starter, Functional Decomposition Diagram

3 Добавьте необходимые формы



4 Составьте структурную карту Константайна

5 Составьте спецификацию программного модуля.

Варианты:

1. Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.
2. Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.
3. Разработать программный модуль «Решение комбинаторно-оптимизационных задач». Модуль должен содержать алгоритмы поиска цикла минимальной длины (задача коммивояжера), поиска кратчайшего пути и поиска минимального связывающего дерева.
4. Разработать приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц, и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.
5. Разработать приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.
6. Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.

7. Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.
8. Разработать программный модуль «Автосервис». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.
9. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
11. Разработать программный модуль «Картотека агентства недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.
12. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.
13. Разработать программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.
14. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.
15. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.
16. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

Примечание. При разработке программы не ограничиваться функциями, приведенными в варианте, добавить несколько своих функций

Контрольные вопросы:

1. Из чего состоит структурная схема?
2. Что такое спецификация программного модуля?
3. В чем заключается методика Константайна?

ЛАБОРАТОРНАЯ РАБОТА № 3 (2 часа)

Тема: Разработка перечня артефактов

Цель: освоение интерфейса программы и навыков построения диаграммы прецедентов, разработка перечня артефактов.

Задание:

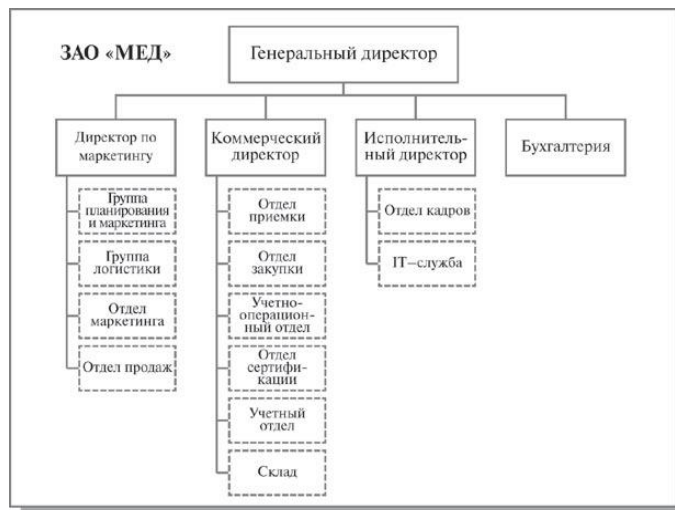
1. Определите внешних исполнителей (контрагентов компании)
2. Создайте «физическую» диаграмму
3. Постройте диаграмму прецедентов

Технология выполнения:

1. Краткое описание предметной области

Компания - дистрибьютор ЗАО "МЕД" закупает медицинские препараты отечественных и зарубежных производителей и реализует их через собственную дистрибьюторскую сеть и сеть аптек. Компания осуществляет доставку товаров, как собственным транспортом, так и с помощью услуг сторонних организаций.

Организационная структура предприятия оптовой торговли ЗАО "МЕД" имеет следующий вид:



Основные цели автоматизации компании "МЕД":

- Разработка и внедрение комплексной автоматизированной системы поддержки логистических процессов компании.
- Повышение эффективности работы всех подразделений компании и обеспечение ведения учета в единой информационной системе.

Основные бизнес-процессы компании - закупки, складирование запасов, продажи, взаиморасчеты с поставщиками и клиентами.

Ключевые функциональные требования к информационной системе:

1. Управление запасами. Оперативное получение информации об остатках на складе.
2. Управление закупками. Планирование закупок в разрезе поставщиков.
3. Управление продажами. Контроль лимита задолженности с возможностью блокировки формирования отгрузочных документов.
4. Полный контроль взаиморасчетов с поставщиками и клиентами.
5. Получение управленческих отчетов в необходимых аналитических срезах - как детальных для менеджеров, так и агрегированных для руководителей подразделений, и директоров фирмы.

Ограничения предметной области

В рамках проекта не рассматривается автоматизация учета основных средств, расчета и начисления заработной платы, управления кадрами. Развертывание новой системы предполагается осуществить только в следующих подразделениях ЗАО "МЕД":

- Отдел закупок;
- Отдел приемки;
- Отдел продаж;
- Отдел маркетинга;
- Группа планирования и маркетинга;
- Группа логистики;
- Учетно-операционный отдел;
- Учетный отдел;
- Отдел сертификации (в части учета сертификатов на медикаменты);

– Бухгалтерия (только в части учета закупок, продаж, поступлений и платежей).

Описание состава автоматизируемых бизнес-процессов

1) Компания "МЕД» осуществляет закупки у отечественных и зарубежных производителей, следовательно, контрагентами компании являются отечественные и зарубежные поставщики медикаментов.

2) Компания пользуется услугами транспортных компаний для доставки медикаментов. Следовательно, внешними контрагентами также являются транспортные компании.

3) Компания реализует медикаменты через дистрибьюторскую сеть и сеть аптек. Следовательно, контрагентами компании являются покупатели (дистрибьюторы, аптеки).

Внешними контрагентами компании "МЕД" являются поставщики (отечественные, зарубежные), покупатели (дистрибьюторы, аптеки) и транспортные компании.

На физической диаграмме компанию изобразим прямоугольником.

Для отображения контрагентов используются графический символ Actor (фигурка человечка).

Для изображения взаимодействия между компанией и внешними контрагентами используются соединительные линии, поименованные для того, чтобы были понятны функции контрагентов по отношению к компании.

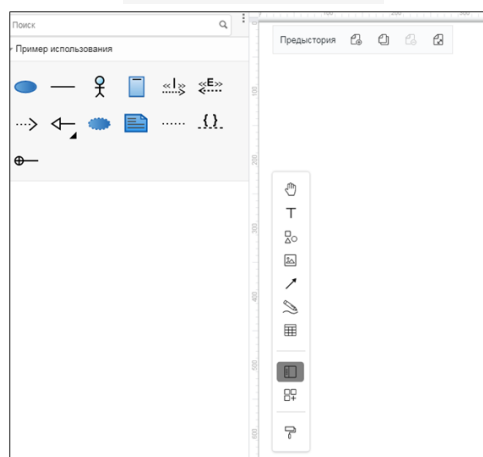
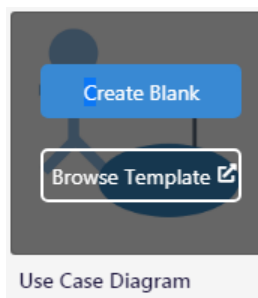
Создание «физической» диаграммы:

1. Запустите Visual Paradigm Online

(<https://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard>)

2. Появится окно, в котором выберите (категорию шаблонов) Вкладку Diagram

3. В открывшемся списке выберите диаграмму Use Case, активизируйте команду Create Blank.



6. Для изображения границ компании «МЕД» выберите из набора графических элементов, представленных в левой части окна, пиктограмму прямоугольника с надписью «Система» и переносите ее на рабочее поле мышкой при нажатой правой клавише, Отрегулируйте размеры прямоугольника.

7. Задайте цвет прямоугольника «Система» и введите подпись «ЗАО «МЕД»».

8. Для изображения на диаграмме контрагентов следует воспользоваться графическим символом с изображением человечка с надписью «Актер» и перенести его на рабочее поле при нажатой правой клавише мышки.

9. Соедините линиями изображение каждого контрагента с прямоугольником используя линии «Ассоциация/Сообщение» и при нажатой левой клавише мышки осуществите соединение фигур.

10. Внесите наименования контрагентов "Покупатели (аптеки)", "Покупатели (дистрибьюторы)", "Поставщики (Россия)", "Поставщики (импорт)", "Транспортные компании". Для того чтобы внести надписи на диаграмме, необходимо на панели инструментов "Форматирование" зафиксировать пиктограмму «Текст» (символ буквы "А"). Щелкните мышкой на изображении человечка, курсор установится на поле с надписью **Актер**. Введите в это поле наименование контрагента.

11. Введите наименование компании "МЕД" в нарисованный прямоугольник, щелкнув мышкой по прямоугольнику. Обратите внимание на то, что при этом должна быть активна пиктограмма «Текст» (символ буквы "А").

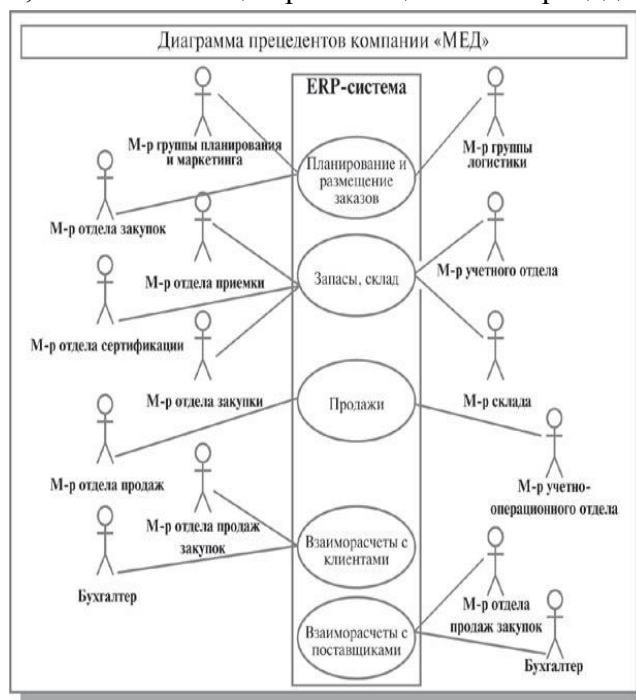
12. Аналогичным образом внесите надписи к линиям соединения фирмы и контрагентов.

Физическая диаграмма ЗАО "МЕД" представлена на рисунке ниже.



2. Построение диаграммы прецедентов

Используя навыки, полученные при выполнении задания 1, постройте диаграмму прецедентов, отображающую прецеденты (варианты использования) компании «Мед» и внутренних исполнителей, обеспечивающих реализацию этих прецедентов внутри системы.



Вариант	Предметная область	Сущность задачи
1	Страховая медицинская компания	Страховая медицинская компания (СМК) заключает договоры добровольного медицинского страхования с населением и договоры с лечебными учреждениями на лечение застрахованных клиентов. При возникновении страхового случая клиент подает заявку на оказание медицинских услуг по условиям договора инспектору, который работает с данным клиентом. Инспектор направляет данного клиента в лечебное учреждение. Отчеты о своей деятельности инспектор предоставляет в бухгалтерию. Бухгалтерия проверяет оплату договоров, перечисляет денежные средства за оказанные услуги лечебным учреждениям, производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики. СМК не только оплачивает лечение застрахованного лица при возникновении с ним страхового случая, но и, при возникновении каких-либо осложнений после лечения, оплачивает лечение этих осложнений.
2	Банк	Банк – это предприятие, осуществляющее регулирование платежного оборота в наличной и безналичной формах. Банк привлекает денежные средства физических и юридических лиц во вклады; размещает привлеченные средства от своего имени и за свой счет; открывает и ведет банковские счета физических и юридических лиц; инкассирует денежные средства, векселя, платежные и расчетные документы; производит кассовое обслуживание физических и юридических лиц; производит куплю-продажу иностранной валюты в наличной и безналичной формах; предоставляет услугу хранения ценных бумаг и драгоценных металлов;
3	Кадровое агентство	Кадровое агентство способствует трудоустройству безработных граждан. Агентство ведет учет и классификацию данных о безработных на основании резюме от них. От предприятий города поступают данные о свободных вакансиях, на основании которых агентство предлагает различные варианты трудоустройства соискателям. В случае положительного исхода поиска вакансия считается заполненной, а безработный становится трудоустроенным. По результатам своей деятельности кадровое агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
4	Компания по разработке программных продуктов	Компания заключает договор с клиентом на разработку программного продукта согласно техническому заданию. После утверждения технического задания определяется состав и объем работ, составляется предварительная смета. На каждый проект назначается ответственный за его выполнение – куратор проекта, который распределяет нагрузку между программистами и следит за выполнением технического задания. Когда программный продукт готов, то его внедряют, производят обучение клиента и осуществляют дальнейшее сопровождение. По результатам своей деятельности компания производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
5	Туроператор	Туроператор предоставляет возможность своим клиентам осуществить туристическую или деловую поездку в различные города России и мира. При разработке нового тура сначала анализируется текущая ситуация на рынке туризма и выбирается направление тура. После этого определяется статус тура, бронируются места в гостиницах и билеты на переезд к месту тура, разрабатывается культурная/деловая/развлекательная программа, утверждаются сроки тура. На каждый тур назначается ответственное лицо от туроператора, которое будет вести данный тур для улаживания проблем в случае возникновения каких-нибудь чрезвычайных или форс- мажорных ситуаций. Клиент приходит в офис туроператора, где вместе с менеджером выбирает уже разработанный тур и оформляет путевку. После возвращения из тура клиент может высказать свои замечания или пожелания, которые будут учтены при доработке существующих туров или при разработке новых. Также, для дальнейшего улучшения тура, туроператор проводит анализ отчетов от посредников (гостиница, гиды и т.д.). По результатам своей деятельности туроператор производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
6	Строительная организация	Строительная организация занимается строительством объектов по заказам клиентов. Сначала заказ проходит предварительную стадию: сбор различных разрешений на строительство, составление эскиза объекта, расчет объема и закупка строительных материалов. Сами строительные материалы доставляются на объект партиями. По мере поступления очередной партии стройматериалов закладывается фундамент объекта, строится каркас здания.

		По результатам данной работы происходит согласование с заказчиком, после чего утепляется контур, вставляются окна, устанавливается крыша. Далее идет обсуждение с клиентом внутренней отделки здания, закупаются отделочные материалы. После того, как объект проходит технический контроль, он передается заказчику. В дополнительные услуги строительной организации входят: услуги дизайнера по интерьеру, закупка и доставка мебели, сотрудничество с охранным предприятием по установке сигнализации. По результатам своей деятельности строительная организация производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
7	Компьютерная компания	Компьютерная компания занимается продажей, ремонтом, сборкой, тестированием компьютерной техники. Также, специалисты компании предоставляют услуги по разработке и монтажу локальных вычислительных сетей. Вся техника и комплектующие закупаются оптом у дилеров и хранятся на складе. Клиент, который хочет приобрести товар, оформляет заказ в торговом зале, а забирает технику со склада или оставляет заявку на ее доставку. Клиент, который хочет отремонтировать технику, приносит ее в сервисный отдел, откуда, по прошествии некоторого времени, забирает как отремонтированную или как технику, не подлежащую ремонту. По желанию клиента, специалисты компании могут выехать к клиенту для общей диагностики возникшей проблемы с техникой. По результатам своей деятельности компьютерная компания производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
8	Компания по предоставлению телекоммуникационных услуг	Компания занимается оказанием телекоммуникационных услуг абонентам. Клиент делает заявку на подключение к телекоммуникационным услугам и ему, по необходимости, устанавливают соответствующее оборудование. Оплата за услуги вносится путем авансовых платежей. Каждый факт предоставления услуги фиксируется соответствующим оборудованием и является основанием для списания соответствующей суммы с личного счета абонента. Клиент в любое время суток может получить отчет об оказанных ему услугах, их стоимости и остатку на личном счете абонента. По результатам своей деятельности компания производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
9	Управляющая компания ЖКХ.	Управляющая компания (УК) ЖКХ занимается обслуживанием жилого фонда города. УК получает финансовые средства от населения и бюджета города в виде компенсаций и субсидий на коммунальные услуги. На основании поступивших средств УК осуществляет текущий ремонт жилого фонда, а также капитальный ремонт согласно плану. Для непосредственного выполнения работ УК нанимает соответствующую рабочую силу (сантехников, дворников, электриков и т.д.). По результатам своей деятельности УКЖКХ производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
10	Автобаза	Автобаза предоставляет услуги по перевозке пассажиров, различных грузов как в черте города, так и между соседними городами. Для регулярных рейсов оплата клиентами услуги происходит в моментах оказания. В остальных случаях клиент должен сделать заявку, которая может быть отклонена. Для междугородных перевозок в диспетчерские автобазы фиксируется маршрут следования рейса. По результатам своей деятельности автобаза производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.
11	Спортивный комплекс	Спортивный комплекс предоставляет услуги по проведению спортивных тренировок. Тренировки, относящиеся к одному виду спорта, объединяются в спортивные секции. Клиент обращается в спортивный комплекс, где получает абонемент на посещение спортивной секции. На основе купленных абонементов составляется расписание тренировок на следующий месяц. Также, в зависимости от загруженности спортивного комплекса, распределяются тренеры спортивных секций. По результатам своей деятельности спортивный комплекс производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.

Контрольные вопросы:

1. Что такое артефакт?
2. Назовите сходства и различия диаграмм прецедентов и «физических» диаграмм?
3. Для чего используются диаграммы прецедентов (вариантов использования)?
4. Что отображает (представляет) «прецедент» на Диаграмме прецедентов?
5. Что такое «эктор» (актер, действующее лицо), что он отображает на диаграмме прецедентов?
6. Какие типы отношений (связей) между экторами и прецедентами используются на диаграммах прецедентов?
7. На какие 3 типа можно подразделять экторов?
8. Что представляет (описывает, отображает) прецедент?
9. Какие типы связей (отношений) допускаются между экторами?

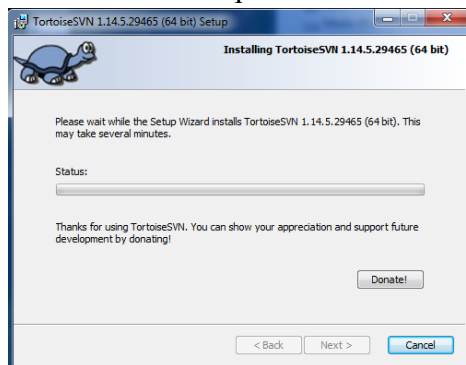
ЛАБОРАТОРНАЯ РАБОТА № 4 (2 часа)

Тема: Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)

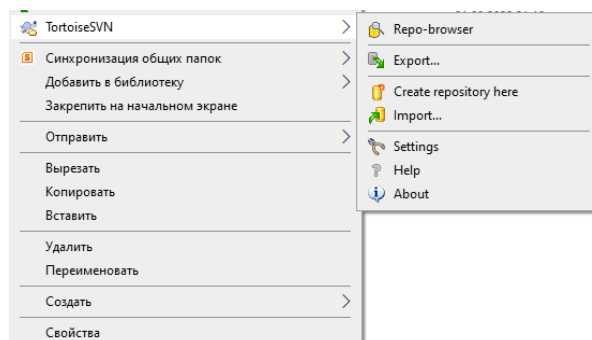
Цель: Изучить на практике понятия и компоненты систем контроля версий (СКВ), приемы работы с ними.

Технология выполнение работы:

1. Установите TortoiseSVN на компьютере.



Сделайте правый клик на папке в проводнике, и вы увидите новые пункты в контекстном меню, такие как эти:

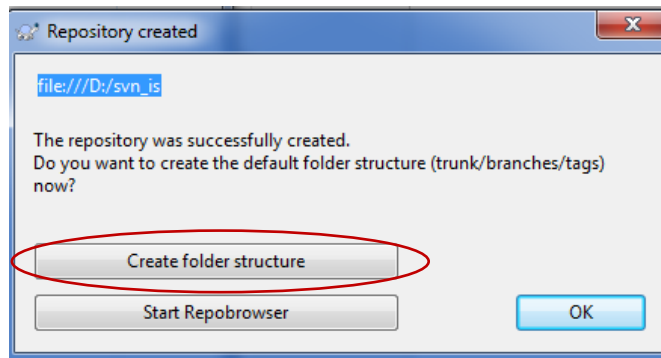


2. Создание хранилища

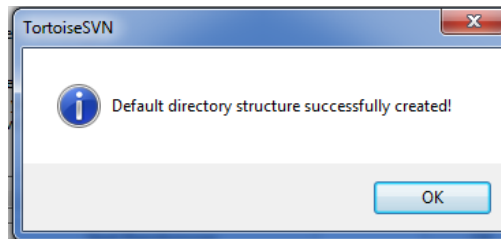
Сначала создайте новую пустую директорию на вашем ПК.

Например, C:\svn_is.

Теперь сделайте правый клик на новой папке и в контекстном меню выберите **TortoiseSVN → Создать здесь хранилище...**



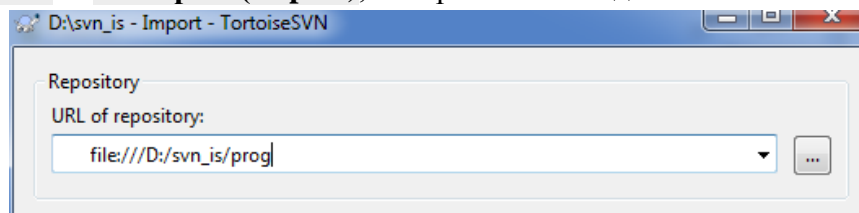
Хранилище, созданное внутри папки, готово к использованию. Создайте внутреннюю структуру папок нажав кнопку ОК.



3. Импорт проекта

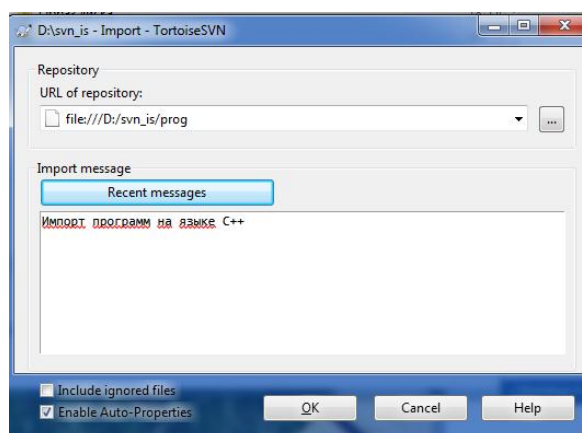
Сейчас у нас есть хранилище, но оно совершенно пустое в данный момент. Добавьте свои файлы в папку **prog** (путь C:\Projects\prog).

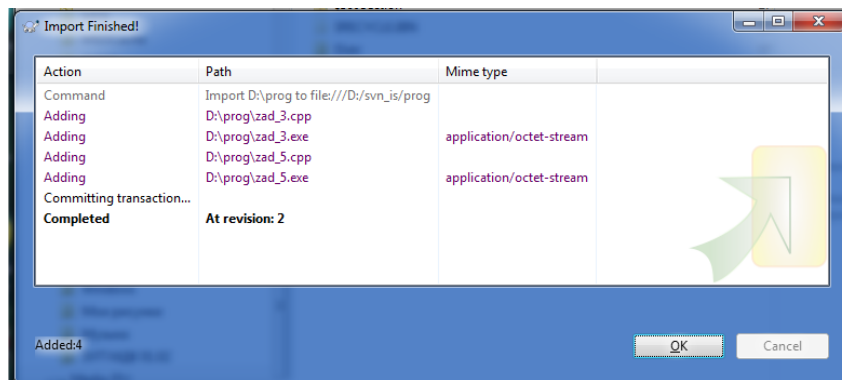
Перейдите к папке **prog** в Проводнике и сделайте правый клик на ней. Теперь выберите пункт **TortoiseSVN** → **Импорт... (Import)**, который вызовет диалог



Другая важная функция данного диалога - это окно **Сообщение импорта (Import messages)**, в которое вы можете добавить сообщение о том, что вы делаете. Когда вам понадобится просмотреть историю проекта, эти сообщения будут ценным подспорьем для просмотра какие изменения и когда были сделаны.

Введите и импортируйте сообщение, например, «Импорт программ на языке C++», затем нажмите ОК.



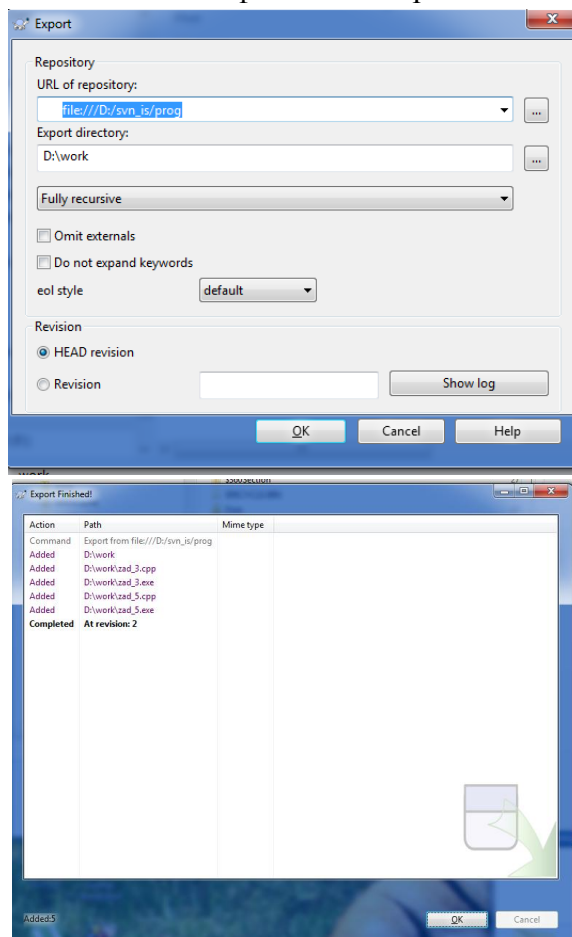


4. Извлечение рабочей копии

В хранилище создайте рабочую копию для повседневной работы. Для создания свежей рабочей копии в Subversion используется термин **Извлечь**.

Извлеките папку **work** из нашего хранилища в папку для разработки называемую C:\Projects\work. Создайте эту папку, затем сделайте правый клик на ней и выберите пункт **TortoiseSVN** → **Извлечь... (Export)**.

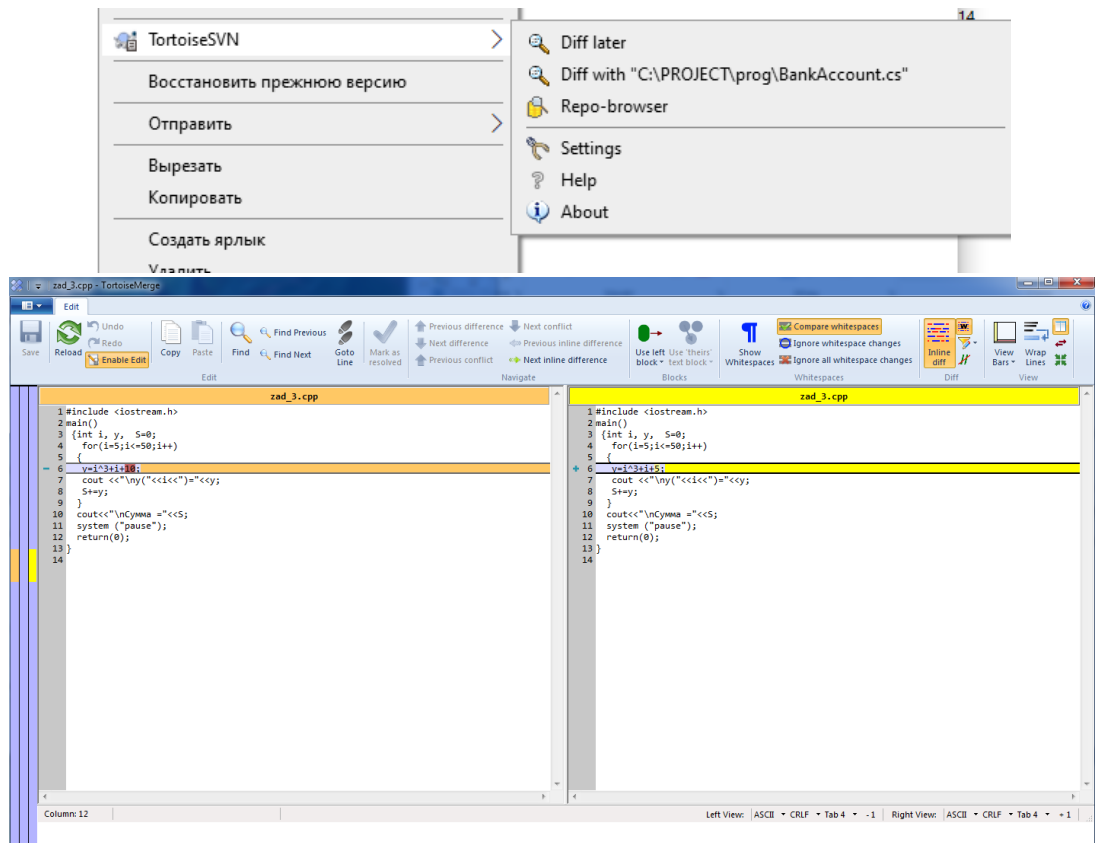
Введите URL-адрес для извлечения, в данном случае **file:///c:/SVN_IS/work** и кликните на ОК. Наша папка для разработки заполнится файлами из хранилища.



5. Внесение изменений

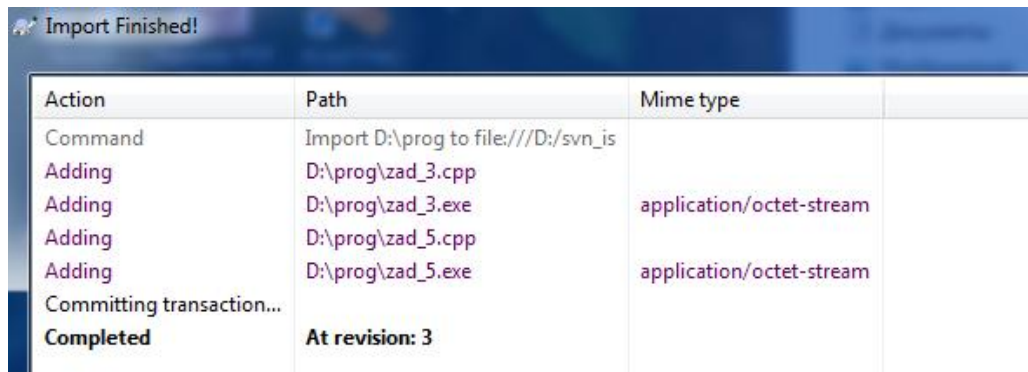
Можно приступить к работе. В папке work измените файлы, например, внесите изменения в файл zad3.c.

Нажмите правой кнопкой на одном из изменённых файлов и выберите команду **TortoiseSVN** → **Различия**. Запустится инструмент TortoiseSVN для сравнения файлов и покажет какие точно строки в файлах были изменены.



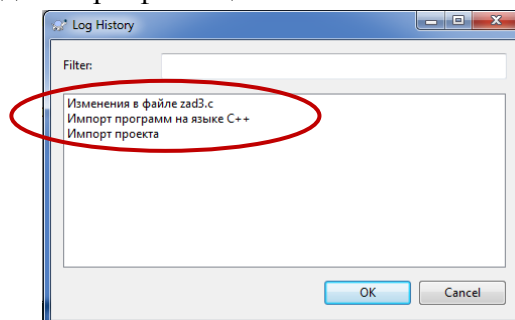
6. Добавление новых файлов

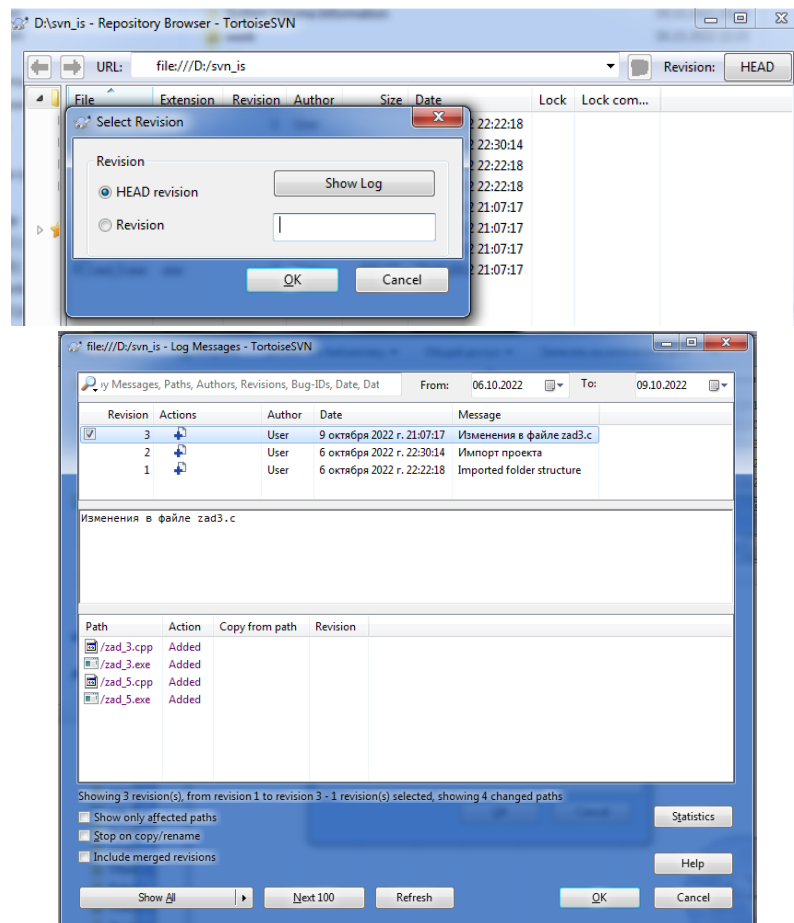
Во время работы над проектом, вам понадобится добавлять новые файлы. Нажмите правой кнопкой на папке и выберите команду **TortoiseSVN** → **Добавить**. Диалог добавления показывает все неверсионированные файлы и вы можете выбрать те файлы, которые вы хотите добавить.



7. Просмотр истории проекта

Одной из самых полезных функций TortoiseSVN является диалоговое окно журнала. Оно показывает список всех фиксаций изменений в файле или папке, а также все те детали сообщения, которые вы вводили при фиксации изменений.





Верхняя панель показывает список всех фиксированных ревизий вместе с началом сообщения фиксации. Если вы выберете одну из этих ревизий, то средняя панель отобразит полное сообщение журнала для той ревизии и нижняя панель покажет список измененных файлов и папок.

8. Отмена изменений

Одной общей функцией всех систем управления ревизиями является функция, которая позволяет вам отменить изменения, которые вы внесли ранее. Если вы хотите избавиться от изменений, которые вы еще не успели фиксировать и восстановить нужный файл в том виде, в котором он был перед началом изменений, то выберите команду **TortoiseSVN** → **Убрать изменения**. Это действие отменит ваши изменения (в Корзину) и вернет фиксированную версию файла, с которой вы начинали. Если же вы хотите убрать лишь некоторых изменения, то вы можете использовать инструмент TortoiseMerge для просмотра изменений и выборочного удаления измененных строк.

Если вы хотите отменить действия определенной ревизии, то начните с диалогового окна журнала и найдите проблемную ревизию. Выберите команду **Контекстное меню** → **Отменить изменения из этой ревизии** и те изменения будут отменены.

Контрольные вопросы:

1. Что такое Subversion?
2. Перечислите возможности TortoiseSVN.
3. Что такое системы контроля версий (СКВ) и для решения каких задач они предназначаются?
4. Объясните следующие понятия СКВ и их отношения: хранилище, commit, история, рабочая копия.
5. Что представляют собой и чем отличаются централизованные и децентрализованные СКВ? Приведите примеры СКВ каждого вида.

ЛАБОРАТОРНАЯ РАБОТА № 5 (2 часа)

Тема: Разработка и интеграция модулей проекта (командная работа)

Цель: изучить принципы разработки и интеграции модулей проекта и принципы формирования и управления командной работы.

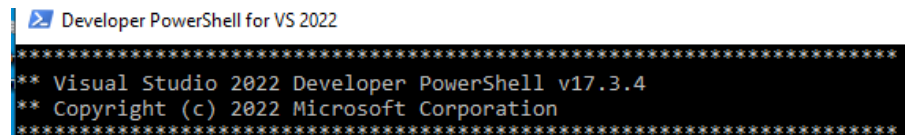
Технология выполнения работы

1. Выполнить поиск командной строки на компьютере двумя способами:

1. Способ - В Windows 10:

1. В поле поиска на панели задач начните вводить имя средства, например dev или developer command prompt. Откроется список установленных приложений, которые соответствуют вашему шаблону поиска.

2. Выберите **Командная строка разработчика для VS 2022** или **PowerShell для разработчиков для VS 2022**.



```
*** Visual Studio 2022 Developer PowerShell v17.3.4 ***
*** Copyright (c) 2022 Microsoft Corporation ***
```

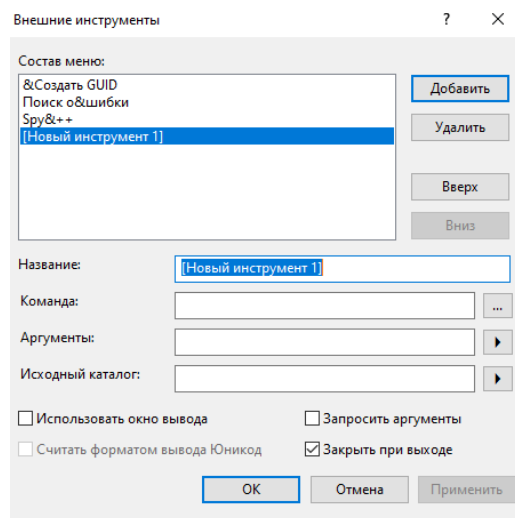
2. Запуск командной строки из Visual Studio.

1. Запустите Visual Studio.

2. Выберите опцию *Продолжить без кода*

3. Выберите меню *Средства* и затем выберите в нем пункт *Внешние инструменты*.

4. В диалоговом окне *Внешние инструменты* нажмите кнопку *Добавить*. Появится новый элемент.



5. Введите название для нового пункта меню, например, Command Prompt.

6. В поле *Команда* укажите файл, который должен запускаться, например, C:\Windows\System32\cmd.exe.

7. В поле *Аргументы* укажите, где найти конкретную командную строку, которую вы хотите использовать, например:

/k "C:\Program Files (x86)\ Microsoft Visual Studio\2017\Enterprise\Common7\Tools\VsDevCmd.bat"

эта команда запускает командную строку разработчика, установленную с Visual Studio 2017 Enterprise). Измените это значение в соответствии с вашей версией, выпуском и расположением установки Visual Studio.

8. Выберите значение для поля *Исходный каталог*, например, Каталог проекта.

9. Нажмите кнопку *ОК*. Новый элемент добавится в меню.

10. Вызовите командную строку из меню *Средства* или в меню *Пуск*

11. Создайте папки project\myProj на диске C:

12. Сделайте текущим папку myProj

Выполните команды:

```
C:\>mkdir project
C:\>cd project
C:\project>mkdir myProj
C:\project>cd myProj
```

13. Создайте файл с помощью команды **Сору** `con prog.cpp`, введите текст

```
#include <iostream>
using namespace std;
int main()
{
    cout <<" Hell, word, from C++! ";
    cout << endl;
}
```

14. Нажмите Ctrl + Z, чтобы сохранить файл, и Ctrl + C, чтобы выйти из редактирования.

15. Введите команду `dir`, чтобы получить список содержимого каталога `C:\project\myProj`

```
Содержимое папки C:\project\myProj
23.10.2022  19:11    <DIR>      .
23.10.2022  19:11    <DIR>      ..
23.10.2022  18:53                298 prog.cpp
                1 файл      298 байт
                2 папок   210 774 020 096 байт свободно
```

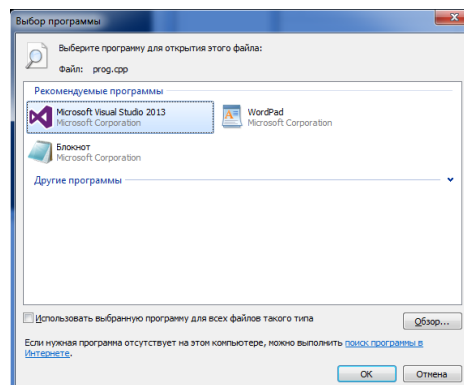
16. Скомпилируйте программу с помощью команды

cl /EHsc prog.cpp

Компилятор `cl.exe` создаст OBJ-файл, содержащий скомпилированный код, а затем запустит компоновщик для создания исполняемой программы с именем `prog.exe`. Это имя отображается в строках информации, выводимой компилятором. Если вы получаете сообщение об ошибке, например, "`cl` не распознается как внутренняя или внешняя команда, исполняемая программа или пакетный файл", ошибке C1034 или LNK1104, командная строка разработчика настроена неправильно.

17. Запустите программу `prog.exe`, если компиляция прошла успешно, набрав в командной строке **prog**

18. Откройте файл выполнив команду `C:\project\myProj>prog.cpp` Выберите в окне программу для открытия файла.



Контрольные вопросы

1. Как происходит управление проектом по стандарту PMBoK?
2. Как с целью структуризации управления проектом распределяются процессы управления проектом?
3. Понятие интеграции процессов управления.
4. Какие главные задачи решаются при организации работы над проектом?
5. Что понимается под командой проекта, как и на какой период она формируется?
6. Дайте характеристику двум основным принципам формирования команды для управления проектом.

ЛАБОРАТОРНАЯ РАБОТА № 6 (4 часа)

Тема: Отладка отдельных модулей программного проекта

Цель: усвоить знание основ модульного программирования; освоить способы создания и применения модулей.

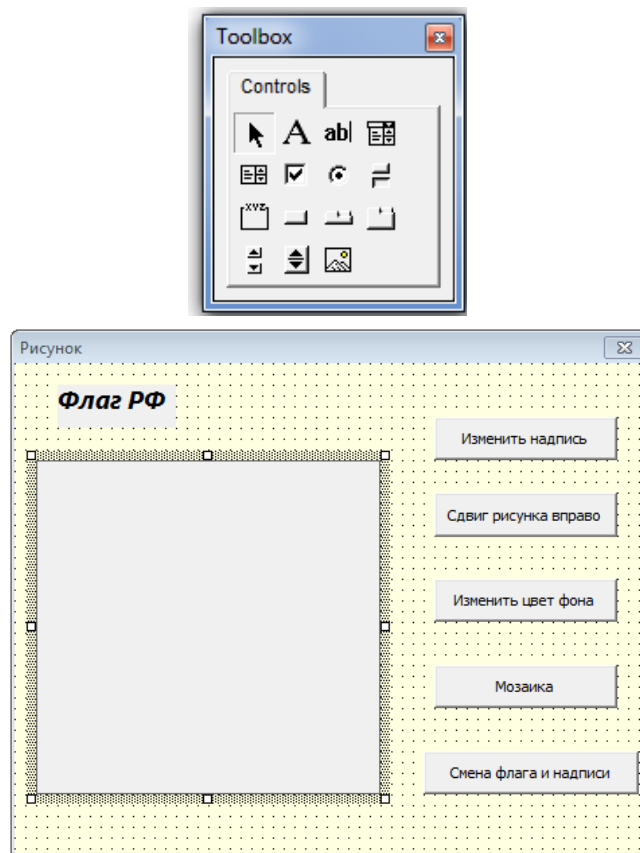
Задание:

1. Провести отладку отдельных модулей программного проекта

Технология выполнения работы:

Задание 1.

1. Откройте новую рабочую книгу
2. Запустите редактор VB
3. Создайте форму пользователя с помощью командой Insert-FormUser, используя инструменты ToolBox и свойства Properties. Добавьте на форму элементы управления: 4 CommandButton, Image и Label.




4. Создайте новую процедуру для кнопки «Измени надпись».
5. Введите текст процедуры. Правильно укажите адрес графического файла. В тексте намеренно сделаем ошибку в свойстве Size (напишем Sie):

```
CommandButton1 Click
Private Sub CommandButton1_Click()
Label1.Caption = "флаг России"
UserForm1.Image1.Picture = LoadPicture("F:\flag-rossiya.jpg")
Label1.Font.Sie = 14
End Sub
```

6. Выполните компиляцию и выведите форму для работы.
7. После появления формы на экране нажмем на кнопку «Измени надпись». Так как в программе заложена ошибка, появится окно сообщения об ошибке, и открывается редактор VBA.

8. Нажмите на кнопку «Debug» (отладка), и отладчик укажет, в какой строке у вас ошибка

```
CommandButton1 Click
Private Sub CommandButton1_Click()
    Label1.Caption = "Флаг России"
    UserForm1.Image1.Picture = LoadPicture("F:\flag-rossiya.jpg")
    Label1.Font.Size = 14
End Sub
```

9. Исправьте ошибку и нажмите на стандартной панели инструментов на кнопку  («Продолжение»).

Тексты программ для кнопок CommandButton2, CommandButton3, CommandButton4, CommandButton5 представлены ниже:

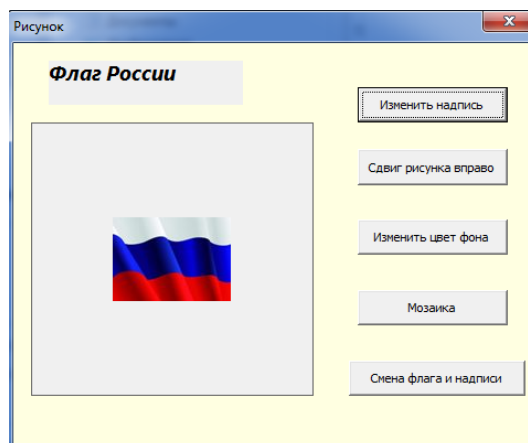
```
Private Sub CommandButton2_Click()
    Image1.PictureAlignment = 4
End Sub

Private Sub CommandButton3_Click()
    Image1.BackColor = &HFF80FF
    UserForm1.BackColor = RGB(64, 0, 0)
End Sub

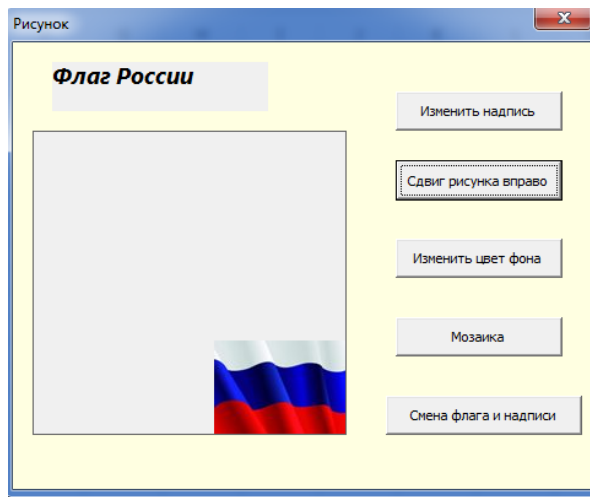
Private Sub CommandButton4_Click()
    Image1.PictureTiling = True
End Sub

Private Sub CommandButton5_Click()
    Label1.Caption = "Флаг Белоруссии"
    Label1.Font.Size = 14
    Label1.Font.Name = "Arial Black"
    UserForm1.Image1.Picture = LoadPicture("F:\флагбел.gif")
End Sub
```

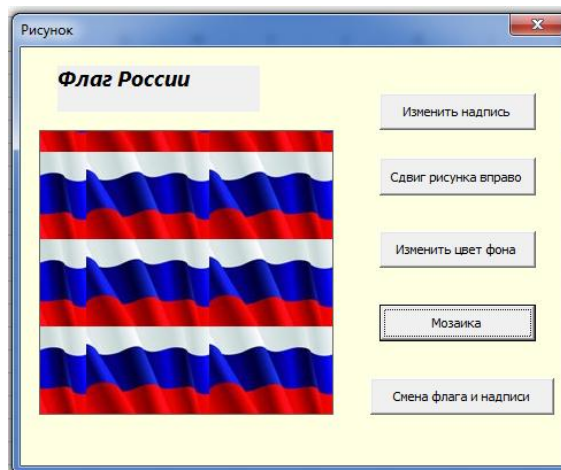
10. После щелчка по кнопке «Измени надпись» форма приобретет вид, представленный на рисунке ниже.



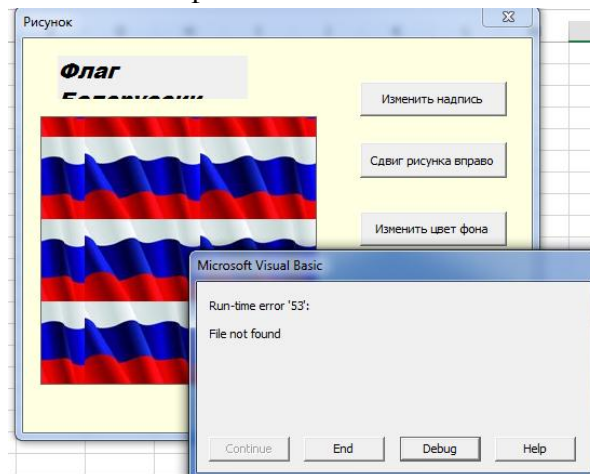
11. После щелчка по кнопке «Сдвинь рисунок вправо» форма приобретет вид, представленный на рисунке.



12. После щелчка по кнопке «Мозаика» форма приобретет вид



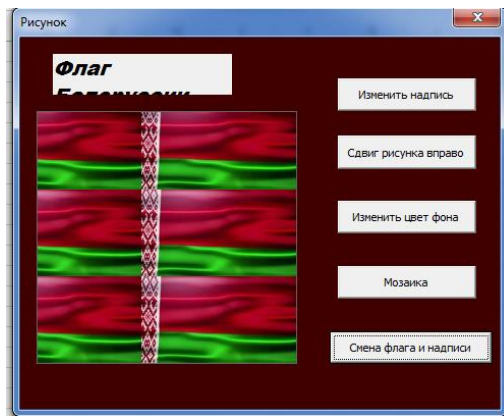
13. После щелчка по кнопке «Смена флага и надписи» появится сообщение об ошибке.



```
Private Sub CommandButton5_Click()
Label1.Caption = "флаг Белоруссии"
Label1.Font.Name = "Arial Black"
UserForm1.Image1.Picture = LoadPicture("F:\флагбел.gif")
End Sub
```

14. Исправьте ошибку в имени файла (расширение jpg)/

```
Private Sub CommandButton5_Click()
Label1.Caption = "флаг Белоруссии"
Label1.Font.Name = "Arial Black"
UserForm1.Image1.Picture = LoadPicture("F:\флагбел.jpg")
End Sub
```



15. Измените размер надписи. Проверьте работу программы.



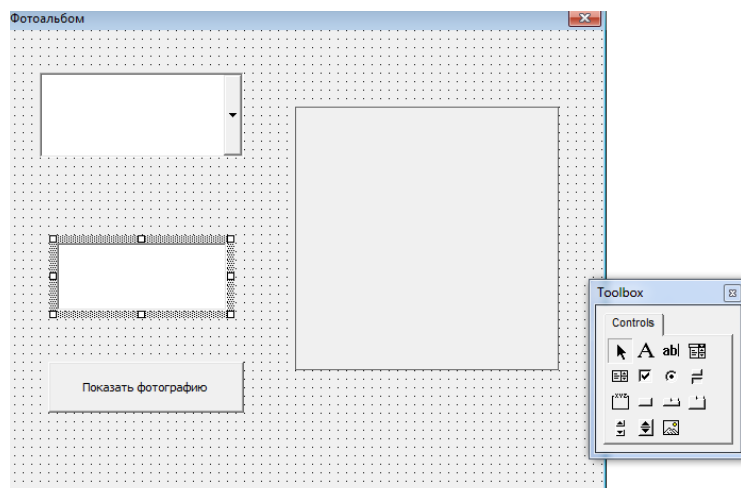
16. Сохраните свою работу.

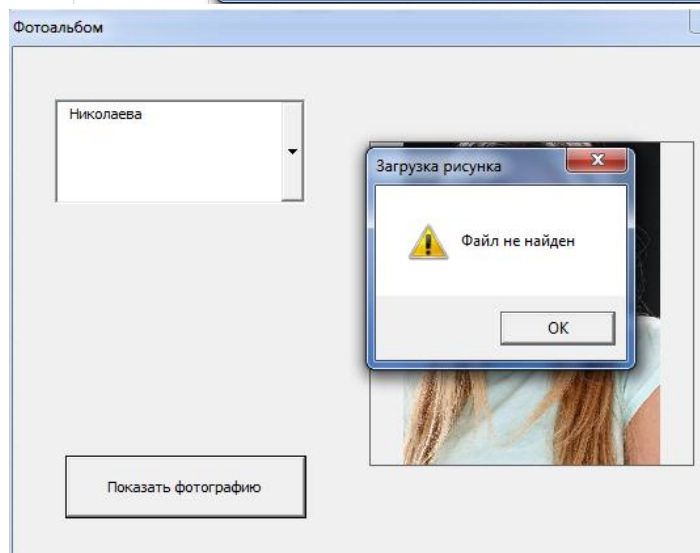
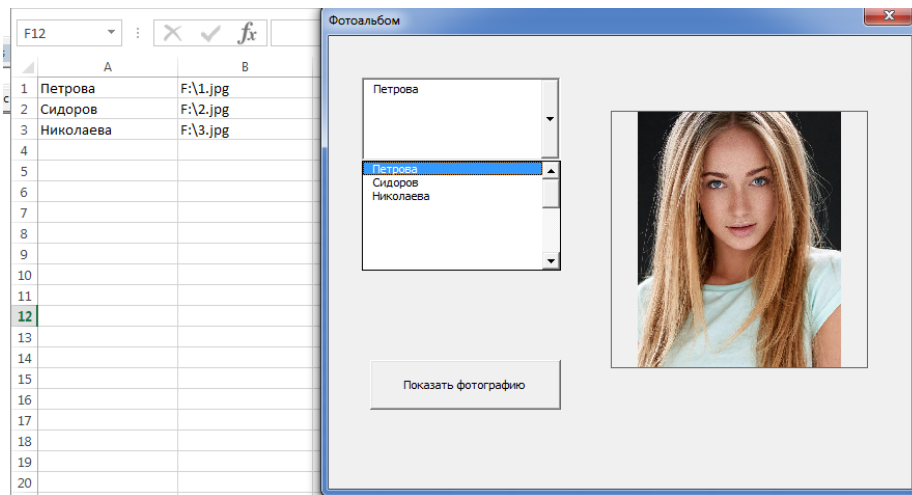
Задание 2.

1. Напишите код на программный продукт с использованием редактора кода VBA.
2. Проведите отладку программного продукта.

Постановка задачи: Создать программу, которая показывает фотографии однокурсников. Фамилии студентов выбираются из раскрывающегося списка ComboBox и при нажатии кнопки «Показать фотографию» фотография выбранного студента отображается в области рисунка.

Для Image1 в свойство PictureSizeMode установить значение fmPictureSizeModeZoom. Фамилии студентов записаны на рабочем листе в столбце A, откуда они заносятся в раскрывающийся список ComboBox. Полные имена файлов, содержащих фотографии записаны в столбце B, откуда они заносятся в список ListBox. Список ListBox сделать невидимым (Enabled=False). При запуске программы раскрывающемся списке должна отображаться фамилия первого в списке студента, а в области рисунка – его фотография. Предусмотреть обработку ошибочных ситуаций. Если имя файла или путь к файлу заданы неверно, вывести об этом сообщение.





Порядок выполнения программы.

Модуль формы состоит из трех процедур:

- процедура обработки ошибочной ситуации;
- процедура обработки события Initialize объекта Userform;
- процедура обработки события CommandButton1_Click.

Процедура Обработка_ошибки с помощью оператора выбора анализирует код ошибки.

Если имя файла задано неверно, при загрузке рисунка возникает ошибка с кодом 53, если путь к файлу задан неверно - ошибка с кодом 76. Обработчик ошибок выводит соответствующее сообщение. При появлении других выполнение программы прерывается, пользователь информируется об ошибке. Осуществляется выход из процедуры.

Процедура обработки ошибочной ситуации

```
Sub Обработка_ошибки()
Select Case Err.Number
Case 53
MsgBox "файл не найден", 48, "Загрузка рисунка"
Case 76
MsgBox "Путь не найден", 48, "Загрузка рисунка"
Case Else
MsgBox "Произошла ошибка: " & Err.Description
End Select
End Sub
```

Процедура обработки события Initialize объекта заполняет список ComboBox из диапазона A1:A20, в который предварительно введены фамилии студентов, список ListBox – из диапазона B1:B20 полными именами файлов. В раскрывающемся списке отображается фамилия первого в списке студента, а в области рисунка - его фотография. При возникновении ошибки

при загрузке рисунка вызывается процедура Обработка_ошибки.

```
UserForm Initialize
Private Sub UserForm_Initialize()
End Sub

Private Sub UserForm_Initialize()
' передача управления на обработчик ошибок,
' помеченный меткой строка1
On Error GoTo строка1
' Список ListBox делаем невидимым
ListBox1.Visible = False
With ListBox1
.ColumnCount = 1
.RowSource = ("B1:B20")
End With
With ComboBox1
.ColumnCount = 1
.RowSource = ("A1:A20")
' Отображение фамилии первого в списке студента
.ListIndex = 0
End With
' Отображение фотографии первого в списке студента
Image1.Picture = LoadPicture(ListBox1.List(0, 0))
Exit Sub
' Обработчик ошибок
строка1:
' вызов процедуры обработки ошибок
Call Обработка_ошибки
End Sub
```

Процедура обработки события **CommandButton1** фотографию выбранного из раскрывающегося списка отображает в области рисунка. Если при загрузке рисунка возникает ошибка, вызывается процедура Обработка_ошибки.

```
Private Sub CommandButton1_Click()
Dim i As Integer
' передача управления на обработчик ошибок,
' помеченный меткой Строка2
On Error GoTo Строка2
i = ComboBox1.ListIndex ' индекс выбранного элемента
If ListBox1.List(i, 0) = "" Then
MsgBox "Не задано имя файла", 48, "Загрузка рисунка"
End If
Image1.Picture = LoadPicture(ListBox1.List(i, 0))
'Выход из процедуры при успешном выполнении загрузки рисунка
Exit Sub
' Обработчик ошибок
Строка2:
' вызов процедуры обработки ошибок
Call Обработка_ошибки
End Sub
```

Контрольные вопросы:

1. Какие ошибки в программах существуют?
2. Что понимают под отладкой программы?
3. Чем отладка отличается от тестирования?

ЛАБОРАТОРНАЯ РАБОТА № 7 (2 часа)

Тема: Организация обработки исключений

Цель: изучить операторы, используемые при обработке исключительных ситуаций, возникающих во время выполнения вычислительных процессов, получить практические навыки в составлении программ.

Технология выполнения работы:

Задание - Оптимизировать программу, производящую простые арифметические операции над числами (сложение, вычитание, умножение и деление), используя обработку исключительных ситуаций (обработка ввода чисел и операции деления).

Задание 1. Реализовать программу простейшего калькулятора

Технология выполнения задания:

- 1 Запустите программу Microsoft Studio
- 2 Создайте консольное приложение на языке C#

```
using System;
namespace ConsoleApplication
{
    class OurClass
    {
        static void Main(string[] args)
        {
            float num1 = 1, num2 = 2, summarize, multiply, sub, divide = 0;
            Console.WriteLine("Введите первое число:");
            try { num1 = float.Parse(Console.ReadLine()); }
            catch
            {
                Console.WriteLine("Неправильный формат числа!\n" +
                    "В качестве значения первого числа будет 1");
            }
            Console.WriteLine("Введите второе число:");
            try { num2 = float.Parse(Console.ReadLine()); }
            catch
            {
                Console.WriteLine("Неправильный формат числа!\n" +
                    "В качестве значения второго числа будет 2");
            }
            summarize = num1 + num2; multiply = num1 * num2; sub = num1 - num2;
            try { divide = num1 / num2; }
            catch (DivideByZeroException)
            {
                Console.WriteLine("Нельзя делить на нуль!");
            }
            Console.WriteLine(
                "\n" + num1 + " + " + num2 + " = " + summarize +
                "\n" + num1 + " * " + num2 + " = " + multiply +
                "\n" + num1 + " - " + num2 + " = " + sub +
                "\n" + num1 + " / " + num2 + " = " + divide);
            Console.WriteLine("\nДля выхода из программы нажмите [Enter]:");
            string anykey = Console.ReadLine();
        }
    }
}
```

```
Введите первое число:5
Введите второе число:6

5 + 6 = 11
5 * 6 = 30
5 - 6 = -1
5 / 6 = 0,8333333

Для выхода из программы нажмите [Enter]:
```

```
Введите первое число:0.45
Неправильный формат числа!
В качестве значения первого числа будет 1
Введите второе число:0.12
Неправильный формат числа!
В качестве значения второго числа будет 2

1 + 2 = 3
1 * 2 = 2
1 - 2 = -1
1 / 2 = 0,5

Для выхода из программы нажмите [Enter]:
```

```

Введите первое число:5
Введите второе число:0

5 + 0 = 5
5 * 0 = 0
5 - 0 = 5
5 / 0 = ?

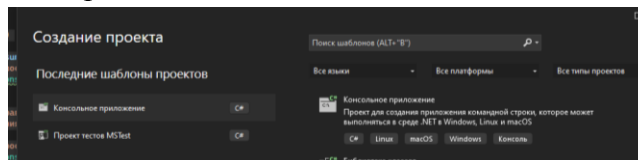
Для выхода из программы нажмите [Enter]:

```

Задание 2. Реализовать программу вычисления периметра четырехугольника

Технология выполнения задания:

- 1 Запустить программу Microsoft Studio
- 2 Создать консольное приложение на языке C#



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            double x1, y1, x2, y2, x3, y3;
            x1 = GetDouble("x1");
            x2 = GetDouble("x2");
            x3 = GetDouble("x3");
            y1 = GetDouble("y1");
            y2 = GetDouble("y2");
            y3 = GetDouble("y3");
            //здесь обработка исключений не требуется, т.к. даже если при вычитании получаются
            //отрицательные числа, то после возведения в квадрат, они будут положительными,
            //т.е. функция корня Sqrt будет всегда обрабатывать только положительное или нулевое значение
            double AB = Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2));
            double BC = Math.Sqrt(Math.Pow(x3 - x2, 2) + Math.Pow(y3 - y2, 2));
            double AC = Math.Sqrt(Math.Pow(x1 - x3, 2) + Math.Pow(y1 - y3, 2));
            double P = AB + BC + AC;
            Console.WriteLine(" периметр = " + P.ToString());
            Console.WriteLine("\nДля выхода из программы нажмите [Enter]:");
            string ankey = Console.ReadLine();
        }

        static double GetDouble(string varname)
        {
            bool IsDouble = false;
            double ReturnValue = 0;

            Console.WriteLine("Введите переменную {0}:", varname);
            while (!IsDouble)
            {
                try
                {
                    ReturnValue = double.Parse(Console.ReadLine());
                    IsDouble = true;
                }
                catch
                {
                    Console.WriteLine("Введенное число не является типом double, попробуйте еще раз!");
                }
            }
            return ReturnValue;
        }
    }
}

```

3 Протестировать программу

```

Введите переменную x1:45
Введите переменную x2:64
Введите переменную x3:67.8
Введенное число не является типом double, попробуйте еще раз!
70
Введите переменную y1:52
Введите переменную y2:58.5
Введенное число не является типом double, попробуйте еще раз!
58
Введите переменную y3:102
периметр = 120,23376490501505
Для выхода из программы нажмите [Enter]:_

```

Контрольные вопросы:

1. Как создается защищенный блок кода?
2. Как описывается процедура обработки конкретного исключения?
3. Как генерируется исключение?
4. Как можно ограничить список исключений, которые могут генерироваться в функции?

ПРАКТИЧЕСКАЯ РАБОТА № 1 (2 часа)

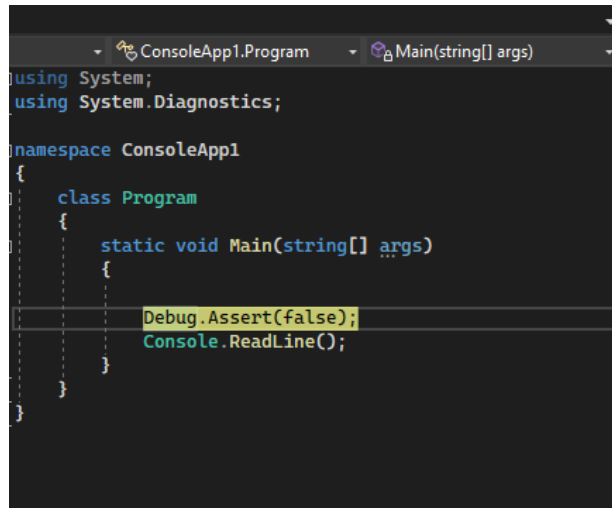
Тема: Применение отладочных классов в проекте

Цель: изучение свойств и методов встроенных классов отладки проектов и применения этих классов для отладки программ

Задание 1 - Создать приложение на C# для отладки кода с использованием Debug.Assert

Технология выполнения работы

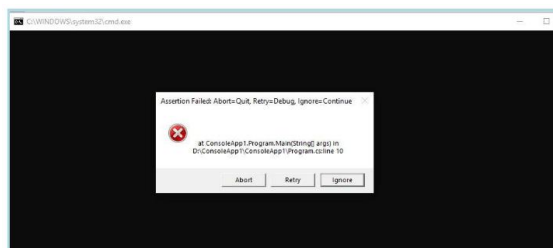
1. Создайте новое консольное приложение C#
2. Введите код программы



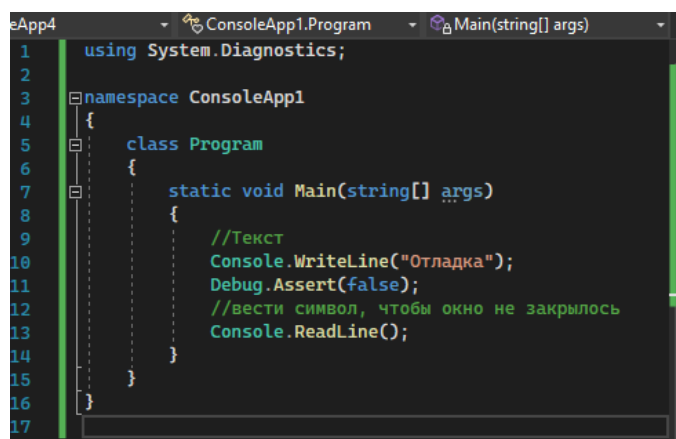
```
using System;
using System.Diagnostics;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Debug.Assert(false);
            Console.ReadLine();
        }
    }
}
```

3. Выполните программу
4. Объясните полученный результат

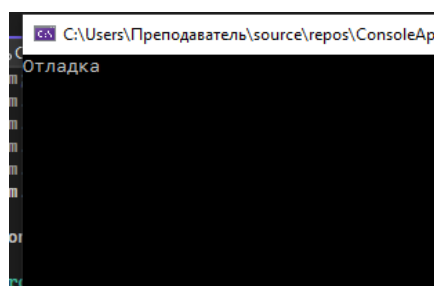


5. Добавьте текст "Отладка"



```
1 using System.Diagnostics;
2
3 namespace ConsoleApp1
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             //Текст
10            Console.WriteLine("Отладка");
11            Debug.Assert(false);
12            //вести символ, чтобы окно не закрылось
13            Console.ReadLine();
14        }
15    }
16 }
17
```

6. Выполните программу.



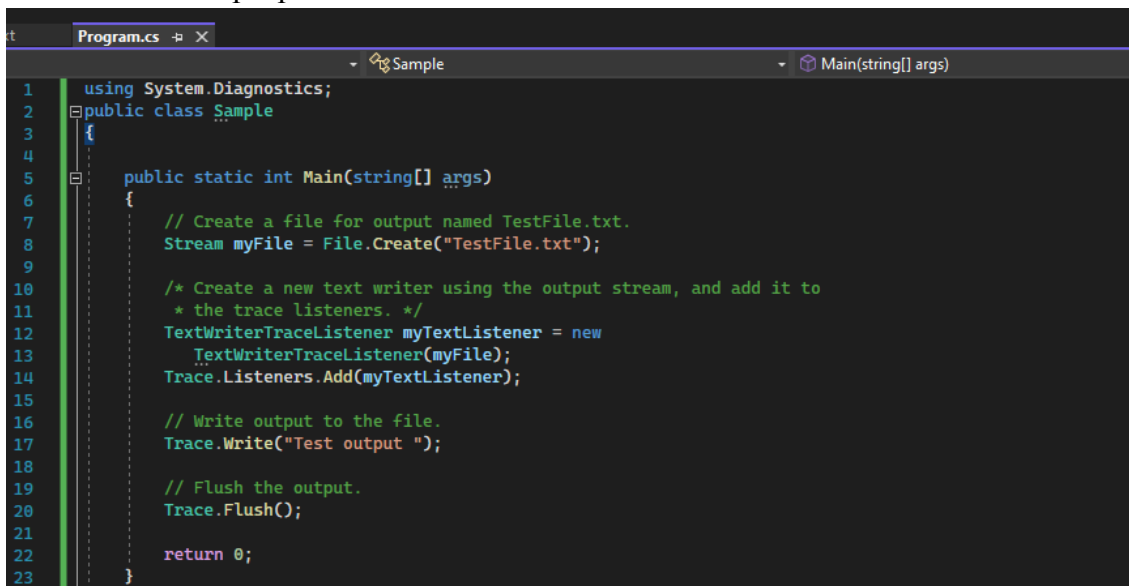
7. Объясните полученный результат
8. Сделайте выводы о работе Debug.Assert

Задание 2 - Рассмотреть пример работы, в котором отладочная информация направляется в разные каналы – окно вывода, консоль, файл

Постановка задачи: Требуется реализовать экземпляр **TextWriterTraceListener** класса, который использует **StreamWriter** вызов **myOutputWriter** для записи в файл с именем **TestFile.txt**. Класс **TextWriterTraceListener** предоставляет **Writer** свойство для получения или задания модуля записи текста, получающего выходные данные трассировки или отладки. Сначала в программе создается файл для вывода. Затем он создает **StreamWriter** первый модуль записи текста, присваивает ему выходной файл и добавляет его в **Listeners**. Затем код выводит одну строку текста в файл. После этого очищается выходной буфер.

Технология выполнения задания:

1. Запустите VS22
2. Создайте консольное приложение C#
3. Введите код программы

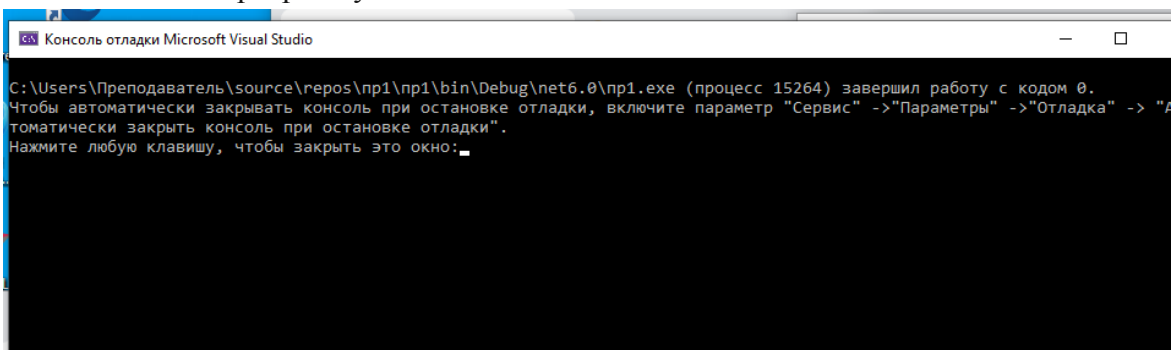


```

1 using System.Diagnostics;
2 public class Sample
3 {
4
5     public static int Main(string[] args)
6     {
7         // Create a file for output named TestFile.txt.
8         Stream myFile = File.Create("TestFile.txt");
9
10        /* Create a new text writer using the output stream, and add it to
11         * the trace listeners. */
12        TextWriterTraceListener myTextListener = new
13            TextWriterTraceListener(myFile);
14        Trace.Listeners.Add(myTextListener);
15
16        // Write output to the file.
17        Trace.Write("Test output ");
18
19        // Flush the output.
20        Trace.Flush();
21
22        return 0;
23    }

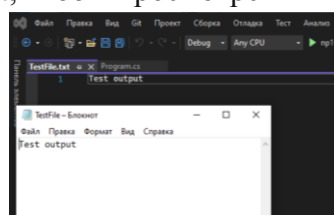
```

4. Выполните программу



Код 0 – Успешное завершение работы программы

5. Откройте папку и проверьте создание файла **TestFile.txt**
6. Откройте файл **TestFile.txt**, чтобы просмотреть выходные данные.



Контрольные вопросы

1. В чем заключается сущность объектно-ориентированного подхода при разработке программного продукта?
2. Какие техники обеспечения качества используются в процессе конструирования?
3. Что такое отладчик?
4. Дайте характеристику свойств и методов классов Debug и Trace.
5. В чем сходство и различие классов Debug и Trace?
6. Основные принципы метода доказательства правильности программ – метода Флойда.
7. Дайте характеристику свойств и методов классов StackTrace и BooleanSwitch.

ЛАБОРАТОРНАЯ РАБОТА № 8 (4 часа)

Тема: Отладка проекта

Цель: изучить действия необходимые для отладки проекта.

Технология выполнения:

Рассмотрим действия для отладки проекта.

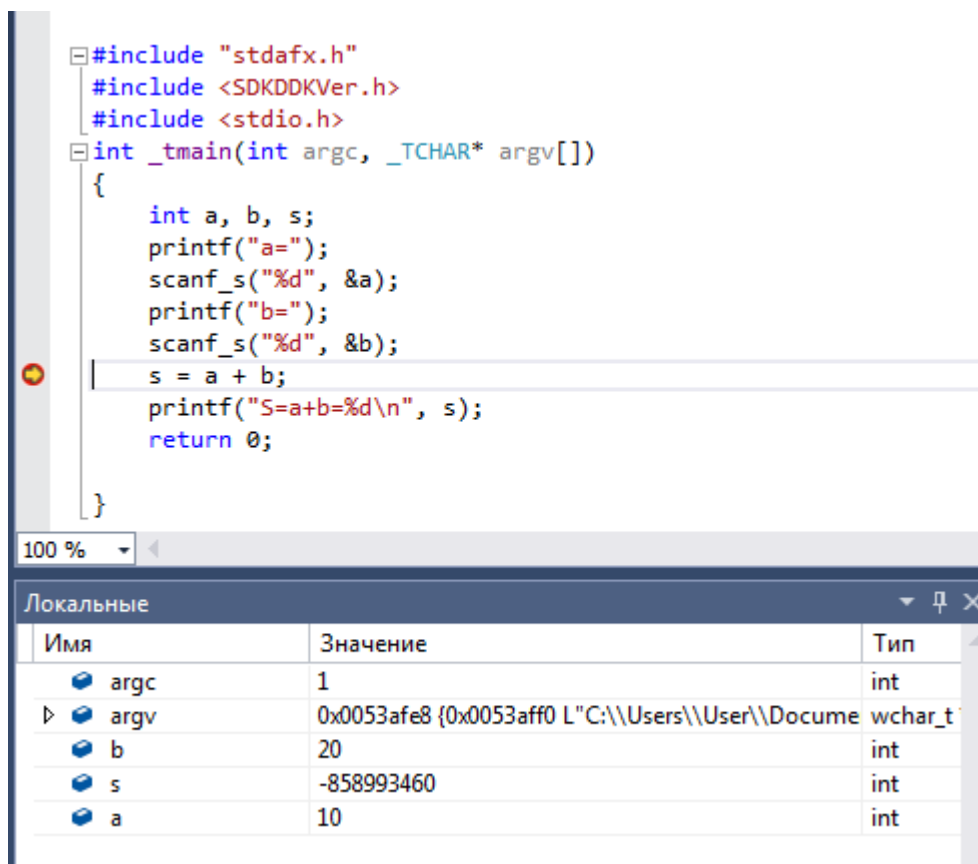
1. Установка точки останова. В файле с исходным кодом программы нужно установить курсор на строчку, где требуется остановка. После этого выполнить одно из действий:

- **Отладка→Точка останова;**
- нажать горячую клавишу F9;
- ПКМ – Вставить точку останова.

```
#include "stdafx.h"
#include <SDKDDKVer.h>
#include <stdio.h>
int _tmain(int argc, _TCHAR* argv[])
{
    int a, b, s;
    printf("a=");
    scanf_s("%d", &a);
    printf("b=");
    scanf_s("%d", &b);
    s = a + b;
    printf("S=a+b=%d\n", s);
    return 0;
}
```

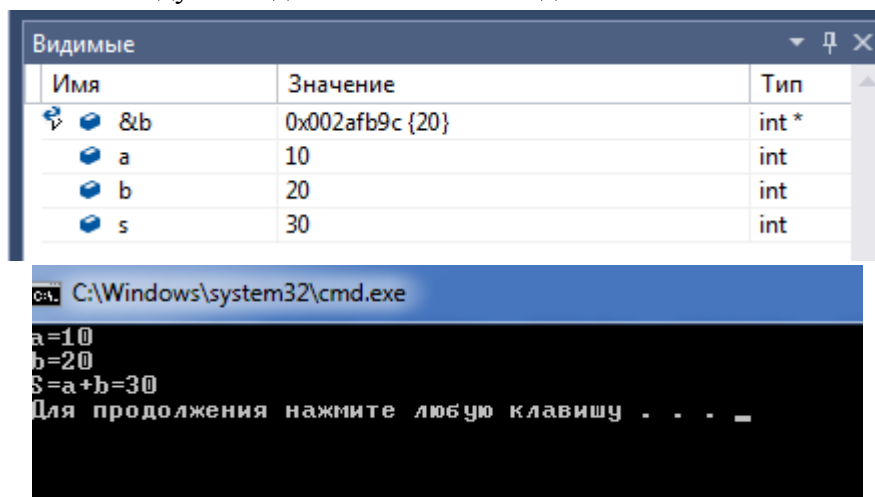
2. Запуск программы до ближайшей точки останова. После указания точки останова можно запустить программу: команда **Отладка→Начать отладку** или нажать горячую клавишу F5.

Программа начнет выполняться до тех пор, пока не будет достигнута точка останова. О достижении этой точки будет свидетельствовать стрелочка, отображаемая поверх точки останова. В этой точке выполнение программы будет остановлено до указания дальнейших действий.



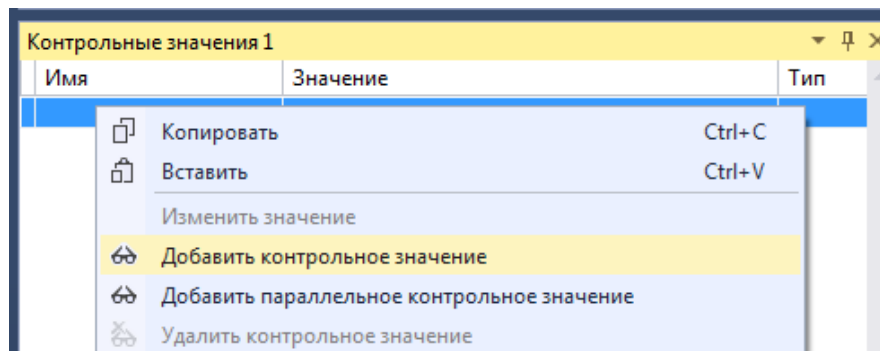
3. Выполнение программы по шагам. Выполнение программы по шагам можно осуществить следующим образом. Команда **Отладка→Шаг с обходом** выполняет одну строчку программы. Если очередным шагом программы является вызов функции, то программист может проанализировать работу этой функции. Для этого в режиме отладки используется команда **Отладка→Шаг с заходом** или горячая клавиша F11. После этой команды отладчик заходит «внутри» функции и программист может продолжить пошаговое выполнение команд.

Если отладка функции закончена и необходимо вернуться в вызывающую программу, то можно использовать команду **Отладка → Шаг с выходом** или сочетание клавиш Shift+F11.

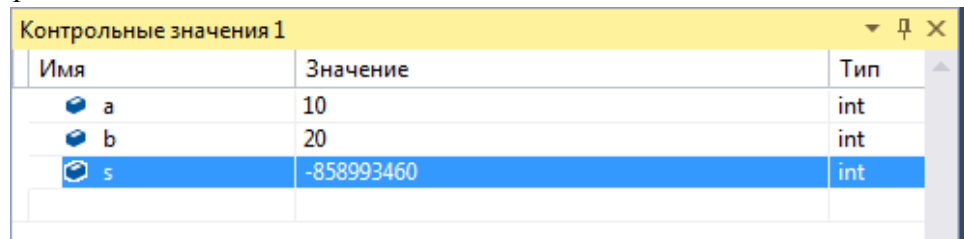


4. Просмотр значения переменных в процессе выполнения программы. В любой момент отладки программист может посмотреть значения переменных, используемых в программе. Для этого используется нижняя часть окна среды Microsoft Visual Studio.

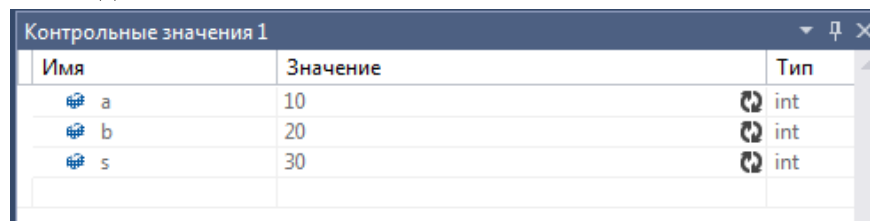
Откройте окно **Контрольные значения**, выбрав **Отладка→Окна →Контрольные значения→Контрольные значения 1**. Вы можете открыть дополнительные окна **контрольных значений**, выбрав окна **2, 3** или **4**.



Для добавления переменной необходимо сделать ее активной и ввести с клавиатуры имя переменной. Если указанная переменная имеется в программе, то ее значение будет автоматически отражено в поле Значение.



Продолжите отладку, выбрав **Отладка → Шаг с заходом** или нажав клавишу **F11** по мере необходимости для перехода. В процессе выполнения значения переменных в окне **Контрольные значения 1** должны меняться.



Таким образом, используя описанные приемы, программист может тщательно проанализировать выполнение написанной программы, проследить изменения значений используемых переменных, устранить некорректную работу реализованных алгоритмов и добиться правильного выполнения программы.

Задание 1 – Отладка программы

Постановка задачи: Вычислить факториал $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$

Технология выполнения:

1. Создайте консольное приложение на C++
2. Введите текст программы

```
(Глобальная область)
#include "stdafx.h"
#include <stdio.h>
int main()
{
    int S=0;
    int i;
    for (i = 2; i <= 20; i++)
        if(i%3!=0) S+=i;
    printf("S=%d\n", S);
    return 0;
}
```

3. Установите точки останова

```

#include "stdafx.h"
#include <stdio.h>
int main()
{
    long int F;
    int i, N;
    printf("N=");
    scanf_s("%d", &N);
    F = 1;
    for (i = 1; i <= N; i++) F = F*i;
    printf("N!=%d\n", F);
    return 0;
}

```

4. Запустите программу до ближайшей точки останова

5. Выполните программу по шагам

6. Добавьте переменные для наблюдения их значений на вкладку **Отладка**→**Окна**

→**Контрольные значения**

Контрольные значения 4		
Имя	Значение	Тип
i	-858993460	int
F	1	long

7. Проанализируйте алгоритм программы и исправьте ошибки.

Задание 2

1 Напишите программу на C++

2 Выполните отладку программы

Варианты заданий:

- Вычислить значение арифметического выражения, заданного формулой $y = \frac{I}{J} \cdot \sin x + I * J$
- Написать программу вычисления объема цилиндра $V = \pi r^2 h$.
- Найдите площадь равнобедренного треугольника по формуле $S = 0,5ab\sqrt{b^2 - \frac{a^2}{4}}$, где a - основание, b-боковая сторона.
- Найдите объем конуса по формуле $V = \frac{\pi R^2 H}{3}$, если известны его радиус основания R и высота H.
- Найдите периметр квадрата по указанному значению его площади, если $P=4a$, $S=a^2$.
- Написать программу вычисления значения функции $y = -2,7x^3 + 0,23x^2 - 1,4$
- Составить программу вычисления объема и площади поверхности куба по данной длине ребра.
- Найти объем шара по формуле $V = \frac{4\pi R^3}{3}$, если известен его радиус.
- Найдите длину окружности и площади круга по указанному радиусу.
- Найдите площадь трапеций по формуле $S = \frac{a+b}{2}h$, если известны его стороны a, b и высота h.
- Вычислить выражение $c = a^2 + b^2 - (2ab/(a-b))$, если известно, что a, b - вещественные числа.
- Найдите силу тока I, если известны сопротивление R и напряжение U по закону Ома: $I = U/R$.
- Вычислите выражение $Z = a \frac{b}{\pi} + 3c$, если известны значения a, b, c.

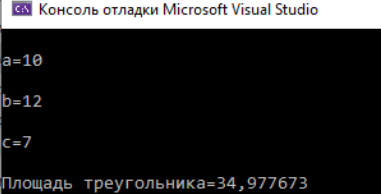
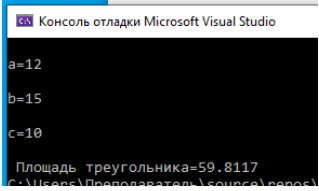
14. Написать программу вычисления значения функции $y = -2,7x^3 + 0,23x^2 - 1,4$

15. Написать программу вычисления значения функции $y = 1,5x^3 - 5x^2 + 0,2$

16. Написать программу вычисления объема пирамиды

Типовой пример 1. Вычислить площадь треугольника по формуле Герона:

$S = \sqrt{p(p-a)(p-b)(p-c)}$, где p — полупериметр треугольника, a, b, c — стороны треугольника.

1 способ	2 способ
<pre>2 // 3 4 #include <iostream> 5 #include <stdio.h> 6 #include <math.h> 7 8 int main() 9 { 10 setlocale(LC_ALL, "Rus"); 11 float a, b, c, p, s; 12 printf("\na="); scanf_s("%f", &a); 13 printf("\nb="); scanf_s("%f", &b); 14 printf("\nc="); scanf_s("%f", &c); 15 p = (a + b + c) / 2; 16 s = sqrt(p * (p - a) * (p - b) * (p - c)); 17 printf("\nПлощадь треугольника=%f", s); 18 } 19</pre> 	<pre>3 4 #include <iostream> 5 #include <stdio.h> 6 #include <math.h> 7 8 int main() 9 { 10 setlocale(LC_ALL, "Rus"); 11 float a, b, c, p, s; 12 std::cout << "\na="; std::cin >> a; 13 std::cout << "\nb="; std::cin >> b; 14 std::cout << "\nc="; std::cin >> c; 15 p = (a + b + c) / 2; 16 s = sqrt(p * (p - a) * (p - b) * (p - c)); 17 std::cout << "\n Площадь треугольника=" << s; 18 } 19</pre> 

Задание 3

1 Напишите разветвленную программу на C++

2 Выполните отладку программы

Варианты заданий:

1. Составить программу нахождения значения функции:

$$y = \begin{cases} x, & \text{если } x < 2; \\ 2, & \text{если } 2 \leq x \leq 3; \\ -x + 5, & \text{если } x > 3. \end{cases}$$

2. Составить программу нахождения значения функции:

$$y = \begin{cases} \operatorname{tg} \frac{1}{\sqrt{x}}, & \text{если } x > 0 \\ \cos \pi x, & \text{если } x \leq 0 \end{cases}$$

3. Составить программу нахождения значения функции:

$$y = \begin{cases} x + 1, & \text{если } x < -1; \\ 1, & \text{если } -1 \leq x \leq 0; \\ x^2, & \text{если } x > 0. \end{cases}$$

4. Составить программу нахождения значения функции:

$$y = \begin{cases} x - 3, & \text{если } x < 1; \\ 4x, & \text{если } 1 \leq x \leq 3; \\ x + 3, & \text{если } x > 3. \end{cases}$$

5. Составить программу нахождения значения функции:

$$y = \begin{cases} 2 - x^2, & \text{если } x > 2 \\ x^3 + 1, & \text{если } x \leq 2 \end{cases}$$

6. Составить программу вычисления выражения:

$$x = \frac{b\sqrt{z+1}}{tz+1}; z = \sqrt{|\sin b|}$$

$$t = \begin{cases} b^2, & b > 1 \\ e^b, & b \leq 1 \end{cases} . \text{ Значение 'b' вводится с клавиатуры.}$$

7. Составить программу нахождения значения функции:

$$y = \begin{cases} x^3, & \text{если } x < 5; \\ 5, & \text{если } 5 \leq x \leq 10; \\ -x + 2, & \text{если } x > 10. \end{cases}$$

8. Составить программу нахождения значения функции:

$$y = \begin{cases} x^2 - 1, & \text{если } x > 1 \\ x + 5, & \text{если } x \leq 1 \end{cases}$$

9. Составить программу нахождения значения функции:

$$y = \begin{cases} x - 1, & \text{если } x \geq 2 \\ x^2 - 4, & \text{если } x < 2 \end{cases}$$

10. Составить программу нахождения значения функции:

$$y = \begin{cases} x^3, & \text{если } x < 0; \\ 10, & \text{если } 0 \leq x \leq 10; \\ -x + x^2, & \text{если } x > 10. \end{cases}$$

Типовой пример 2. Программа вычисления кусочно-непрерывной функции:

$$y = \begin{cases} x^3 - 3, & \text{если } x > 1 \\ 0.5, & \text{если } x = 1 \\ \sqrt{1-x}, & \text{если } x < 1 \end{cases}$$

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <math.h>
int main()
{
    int x;
    float y;
    printf("\nx=");
    scanf_s("%d", &x);
    if (x > 1) y = pow(x,3)-3;
    else
    {
        if (x == 1) y = 0.5;
        else y = sqrtf(1 - x);
    }
    printf("y=%f", y);
    return 0;
}
```

Контрольные значения 1

Имя	Значение	Тип
x	5	int
y	122.000000	float

Контрольные значения 1

Имя	Значение	Тип
x	1	int
y	0.500000000	float

Контрольные значения 1		
Имя	Значение	Тип
x	-3	int
y	2.00000000	float

Задание 4

- 1 Напишите циклическую программу на C++
- 2 Выполните отладку программы

Варианты заданий:

1. Написать программу для подсчета суммы чисел, кратных 3 в диапазоне от 30 до 60.
2. Написать программу для подсчета произведения четных чисел в диапазоне от 1 до 20.
3. Написать программу для подсчета суммы нечетных чисел в диапазоне от 0 до 30.
4. Написать программу для подсчета произведения, некратных 5 чисел в диапазоне от 3 до 30.
5. Написать программу для подсчета суммы чисел, кратных 7 в диапазоне от 10 до 50.
6. Написать программу для подсчета суммы чисел, некратных 4 в диапазоне от 0 до 40.
7. Написать программу для подсчета произведения, кратных 4 чисел в диапазоне от 30 до 50.
8. Написать программу для подсчета произведения, кратных 5 чисел в диапазоне от 20 до 40.
9. Написать программу для подсчета суммы N чисел.
10. Написать программу для подсчета произведения N чисел.
11. Написать программу для подсчета произведения четных чисел в диапазоне от 20 до 45.
12. Написать программу для подсчета суммы чисел, кратных 6 в диапазоне от 15 до 35.
13. Написать программу для подсчета суммы чисел, некратных 6 в диапазоне от 12 до 30.
14. Написать программу для подсчета произведения нечетных чисел в диапазоне от 25 до 40.
15. Написать программу для подсчета суммы чисел, кратных 2 в диапазоне от 10 до 50.
16. Написать программу для подсчета произведения, некратных 2 чисел в диапазоне от 20 до 30.

Замечания:

1. Проверка кратности N для переменной i **if (i% N == 0) ...;**
2. Проверка не кратности N для переменной I **if (i% N != 0) ...;**

Типовой пример 3. Программа для подсчета суммы чисел, некратных 3 в диапазоне от 2 до 20.

```
#include "stdafx.h"
#include <stdio.h>
int main()
{
    int S=0;
    int i;
    for (i = 2; i <= 20; i++)
        if(i%3!=0) S+=i;
    printf("S=%d\n", S);
    return 0;
}
```

Контрольные вопросы

1. Что включает в себя компиляция проекта?
2. Как найти ошибку в тексте программы, если она была обнаружена компилятором?
3. Что такое отладка?
4. Какие возможности отладки проекта предоставляет среда Microsoft Visual Studio?
5. Каким образом можно просмотреть значения переменных при отладке проекта?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2 (2 часа)

Тема: Инспекция кода модулей проекта

Цель: изучение формальных инспекций программного кода.

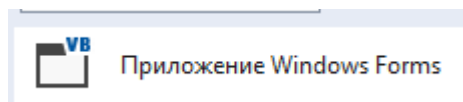
Задание. Написать программу, которая запрашивает у нового сотрудника имя, фамилию и дату рождения. Требуется создать новый класс с именем **Person** для хранения этой информации в свойствах, и создать метод класса, который будет вычислять текущий возраст нового сотрудника.

Добавление в ваш проект нового класса

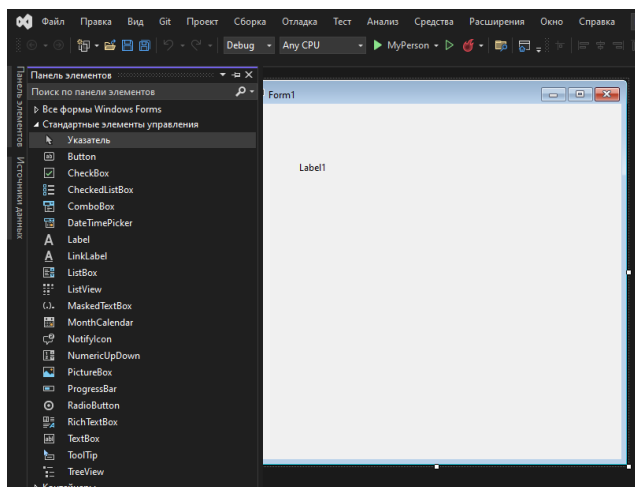
Класс, определенный пользователем, позволяет определить в программе ваши собственные объекты, которые имеют свойства, методы и события, точно так же, как объекты, создаваемые на формах Windows с помощью элементов управления из Области элементов. Чтобы добавить в ваш проект новый класс, щелкните в меню *Проект* на команде *Добавить класс*, а затем определите этот класс с помощью кода программы и нескольких новых ключевых слов Visual Basic.

Создание проекта

1. Запустите Visual Studio, затем создайте в своей папке новый проект с именем **Сотрудники**.



2. Используйте элемент управления *Label* и добавьте в верхней части формы Form1 метку.



3. Используйте элемент управления *TextBox* и нарисуйте под меткой три текстового поля.

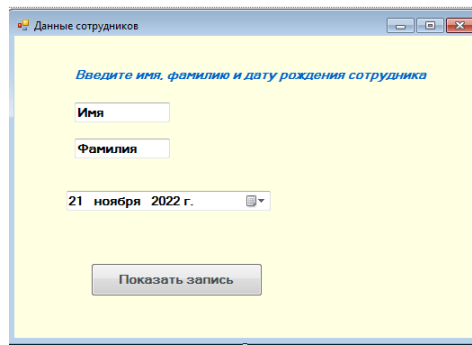
4. Используйте элемент управления *DateTimePicker* и создайте под текстовыми полями объект выбора даты и времени.

5. Используйте элемент управления *Button* и вставьте под объектом выбора даты и времени кнопку.

6. Установите для объектов формы следующие свойства:

Объект	Свойство	Установка
Label1	Text	Введите имя, фамилию и дату рождения сотрудника.
TextBox1	Text	Имя
TextBox2	Text	Фамилия
Button1	Text	Показать запись
Form1	Text	Данные сотрудников

7. Ваша форма должна выглядеть примерно так.

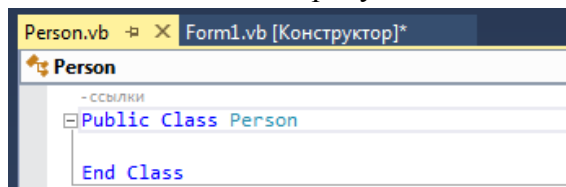


Это базовый интерфейс пользователя для формы, которая определяет запись нового сотрудника фирмы. (Эта форма не подключена к базе данных, так что храниться может только одна запись.) Теперь вы должны добавить в проект класс для хранения информации из этой записи.

8. Щелкните на команде *Добавить класс* в меню *Проект*.

Диалоговое окно *Добавление нового элемента* дает возможность задать имя вашего класса. Когда вы присвоите имя, обратите внимание, что вы можете сохранить в новом модуле класса несколько классов и указать имя, которое будет для них общим.

9. Введите в текстовом поле *Имя* имя *Person.vb*, а затем щелкните *Добавить*. *Visual Studio* откроет в Редакторе кода пустой модуль класса и добавит имя файла *Person.vb* в ваш проект в *Обозревателе решений*, как показано на рисунке.



Объявление переменных класса

Под оператором программы `Public Class Person` введите следующие объявления переменных:

Private Name1 As String

Private Name2 As String

Здесь вы объявляете две переменные, которые будут использованы исключительно в модуле класса для хранения значений двух строковых свойств. Переменные объявлены с помощью ключевого слова `Private`, так как по соглашению Visual Basic программисты должны держать внутренние переменные класса закрытыми - другими словами, недоступными для просмотра извне самого модуля класса.

Создание свойств

1. Под объявлением переменных введите следующий оператор программы и нажмите клавишу (Enter):

Public Property FirstName() As String

Этот оператор создает свойство вашего класса с именем `FirstName`, которое имеет тип `String`. Когда вы нажмете (Enter), VisualStudio немедленно создаст структуру кода для остальных элементов объявления свойства. Требуемыми элементами являются: блок `Get`, который определяет, что программисты увидят, когда будут проверять свойство `FirstName`, блок `Set`, который определяет, что произойдет, когда свойство `FirstName` будет установлено или изменено, и оператор `EndProperty`, который отмечает конец процедуры свойства.

2. Заполните структуру процедуры свойства `FirstName`

```

Public Property FirstName() As String
  Get
    Return Name1
  End Get
  Set (ByVal Value As String)
    Name1 = Value
  End Set
End Property

```

Ключевое слово `Return` указывает, что при обращении к свойству `FirstName` будет возвращена строковая переменная `Name1`. При установке значения свойства блок `Set` присваивает переменной `Name1` строковое значение. Обратите особое внимание на переменную `Value`, используемую в процедурах свойств для обозначения значения, которое присваивается свойству класса при его установке. Хотя этот синтаксис может выглядеть странно, просто поверьте мне - именно так создаются свойства в элементах управления, хотя более сложные свойства будут иметь здесь дополнительную программную логику, которая будет проверять значения и производить вычисления.

3. Под оператором `End Property` введите для свойства `LastName` вашего класса вторую процедуру свойства.

```

Public Property LastName() As String
  Get
    Return Name2
  End Get
  Set (ByVal Value As String)
    Name2 = Value
  End Set
End Property

```

Эта процедура свойства аналогична первой, за исключением того, что она использует вторую строковую переменную (`Name2`), которую вы объявили в верхней части кода класса. Вы закончили определять два свойства вашего класса. Теперь перейдем к методу с именем `Age`, который будет определять текущий возраст нового сотрудника на основе даты рождения.

Создание метода

Под процедурой свойства `LastName` введите следующее определение функции:

```

Public Function Age (ByVal Birthday As Date) As Integer
  Return Int (Now.Subtract(Birthday).Days / 365.25)
End Function

```

Чтобы создать метод класса, который выполняет некое действие, добавьте в ваш класс процедуру `Sub`. Хотя многие методы не требуют для выполнения своей работы аргументов, метод `Age`, определенный мной, требует для своих вычислений аргумент `Birthday` типа `Date`. Этот метод использует для вычитания даты рождения нового сотрудника из текущей системной даты метод `Subtract`, и возвращает значение, выраженное в днях, деленных на 365.25 - примерную длину одного года в днях. Функция `Int` преобразует это значение в целое, и это число с помощью оператора `Return` возвращается в вызывающую процедуру - как и в случае с обычной функцией.

Определение класса закончено! Вернитесь к форме `Form1` и используйте новый класс в процедуре события.

```

Public Class Person
    Private Name1 As String
    Private Name2 As String

    ссылка 2
    Public Property FirstName() As String
        Get
            Return Name1
        End Get
        Set(ByVal Value As String)
            Name1 = Value
        End Set
    End Property

    ссылка 1
    Public Property LastName() As String
        Get
            Return Name2
        End Get
        Set(ByVal Value As String)
            Name2 = Value
        End Set
    End Property

    ссылка 0
    Public Function Age(ByVal Birthday As Date) As Integer
        Return Int(Now.Subtract(Birthday).Days / 365.25)
    End Function

```

Создание объекта с помощью нового класса

1. Щелкните в Обозревателе решений на значке Form1.vb. Появится интерфейс пользователя Form1.
2. Чтобы открыть в Редакторе кода процедуру события Button1_Click, сделайте двойной щелчок мышью на кнопке **Показать запись**.

```

Form1.vb*  Person.vb*  Form1.vb [Конструктор]*
Button1 Click
ссылка 2
Public Class Form1
    ссылка 0
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    End Sub
End Class

```

3. Введите следующие операторы программы:

```

Public Class Form1
    ссылка 0
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim Employee As New Person
        Dim DOB As Date
        Employee.FirstName = TextBox1.Text
        Employee.LastName = TextBox2.Text
        DOB = DateTimePicker1.Value.Date
        MsgBox(Employee.FirstName & " " & Employee.LastName _
            & "в возрасте" & Employee.Age(DOB) & "лет.")
    End Sub
End Class

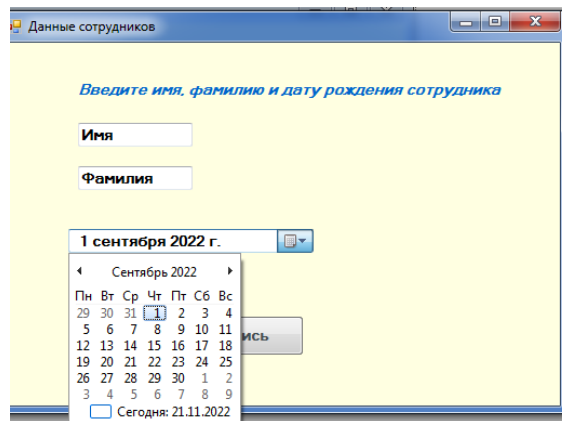
```

Эта процедура сохраняет в объекте с именем **Employee**, который имеет тип **Person**, значения, введенные пользователем. Ключевое слово **New** указывает, что вы хотите немедленно создать новый экземпляр объекта Employee. Теперь нужно объявить переменную с помощью класса, созданного вами самими! Затем процедура объявляет переменную с именем DOB типа Date. Она будет хранить дату, введенную пользователем, и устанавливает свойства FirstName и LastName объекта Employee равными имени и фамилии, введенным в два объекта текстовых полей формы. Значение, возвращаемое объектом выбора даты и времени, сохраняется в переменной DOB, а последний оператор программы отображает окно сообщения, содержащее свойства FirstName и LastName, а также возраст нового сотрудника, определенный методом Age, который при передаче в него переменной DOB возвращает целое значение. Как только вы определили класс в модуле класса, его легко можно использовать в процедуре события.

4. Чтобы запустить программу, щелкните на кнопке **Начать отладку** (F5). В среде разработки появится интерфейс пользователя, готовый к приему ваших данных.

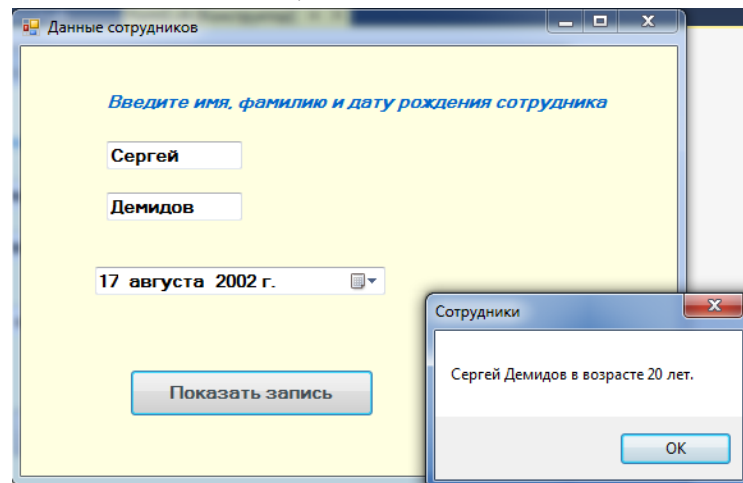
5. Введите в текстовое поле `FirstName` ваше имя, а в текстовое поле `LastName` - фамилию.

6. Щелкните на раскрывающемся списке объекта выбора даты и времени, и прокрутите его до вашей даты рождения.



7. Щелкните на кнопке *Показать запись*. Ваша программа сохраняет значения имени и фамилии в свойствах и использует метод *Age* для вычисления текущего возраста нового сотрудника. Появится диалоговое окно с результатом.

8. Чтобы закрыть это окно сообщения, щелкните на *ОК*, а затем поэкспериментируйте с несколькими различными значениями дат, щелкая на *Показать запись*



Контрольные вопросы:

1. Что такое инспекция кода?
2. Какие виды инспекции кода вы знаете?
3. Определите понятия класс, экземпляр класса, объект.
4. Краткая характеристика метода экспертных исследований программного кода или документации на корректность или непротиворечивость.
5. Дайте характеристику этапов формальной инспекции и ролей её участников.
6. В чём заключается сущность неформальной инспекции?

ЛАБОРАТОРНАЯ РАБОТА 9 (2 часа)

Тема: Тестирование интерфейса пользователя средствами инструментальной среды разработки

Цель: изучения процесса тестирования интерфейса пользователя различными методами.

Технология выполнения работы:

1. Определите предметную область и сферу применения программного продукта.
2. Определите целевую аудиторию.
3. Постройте описательную модель пользователя (профиль). При необходимости — выделите группы пользователей.

4. Сформируйте сценарий поведения пользователей для ручного тестирования пользовательского интерфейса
5. Проведите тестирование интерфейса пользователя.

Контрольные вопросы

1. Что такое интерфейс?
2. Назовите цели тестирования и особенности тестирования при тестировании графического интерфейса пользователя (GUI).
3. Назовите типы требований к интерфейсу пользователя.
4. Назовите этапы функционального тестирования пользовательского интерфейса.
5. Назовите виды покрытий интерфейса пользователя (UI).
6. Достоинства и недостатки ручного тестирования.
7. Достоинства и недостатки автоматического тестирования.
8. Метод поиска элементов UI с использованием распознавания образов и (или) сравнение с образцом.
9. В чём сущность координатного метода тестирования UI?
10. Дайте характеристику Accessibility-метода.

ЛАБОРАТОРНАЯ РАБОТА № 10 (2 часа)

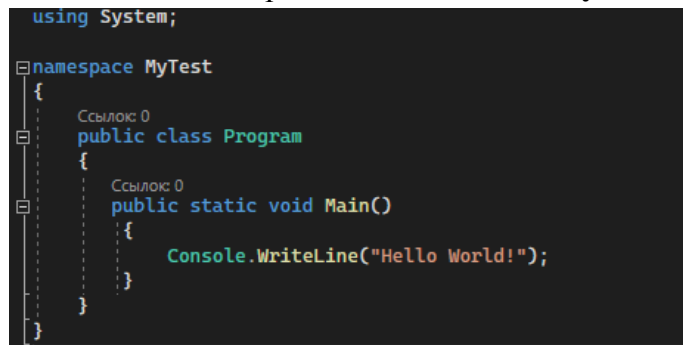
Тема: Разработка тестовых модулей проекта для тестирования отдельных модулей

Цель: научиться выполнять модульное тестирование программного продукта средствами Visual Studio.

Задание 1 - Создание модульных тестов

Технология выполнения работы:

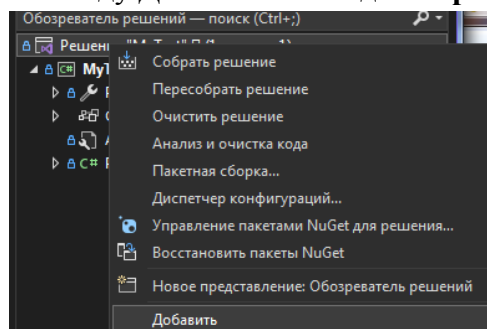
1. Запустите Visual Studio
2. Создайте простой консольный проект C# с именем **MyTest**.



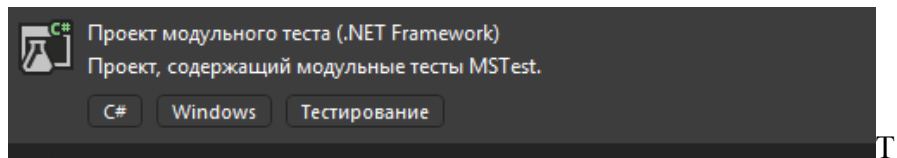
```
using System;

namespace MyTest
{
    Ссылка: 0
    public class Program
    {
        Ссылка: 0
        public static void Main()
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

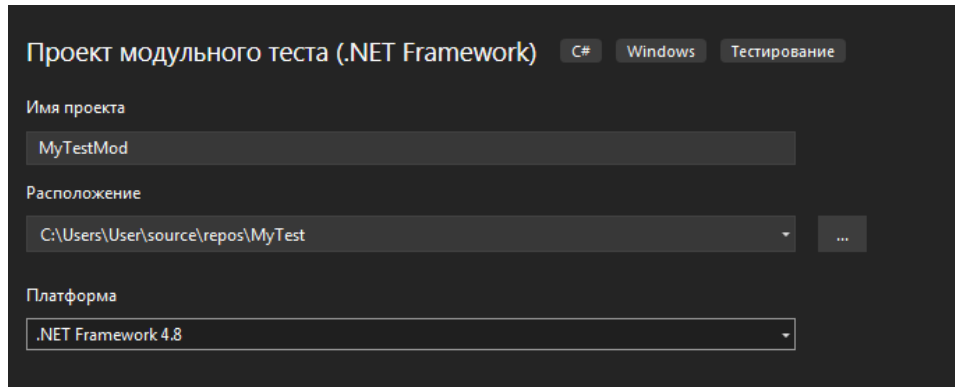
3. Выберите узел решения в **Обзревателе решений**. Затем с помощью щелкнув правой кнопкой мыши активизируйте команду **Добавить** → **Создать проект...**



4. В диалоговом окне нового проекта найдите проект модульного теста, который хотите использовать. Введите **Тестирование** в поле поиска, чтобы найти шаблон проекта модульного теста для тестовой среды C#.

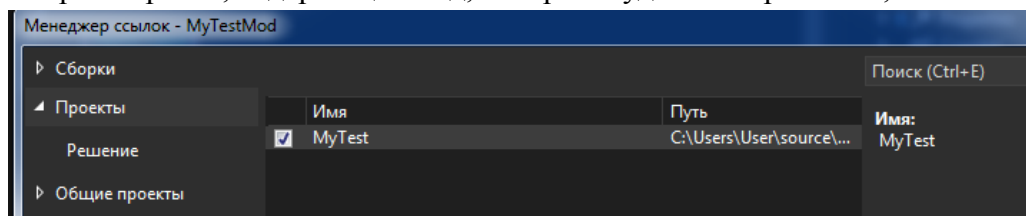


5. Нажмите **Далее**, выберите имя для тестового проекта и нажмите **Создать**.



6. В проекте модульного тестирования добавьте ссылку на проект, который вы хотите протестировать, щелкнув правой кнопкой мыши **Ссылки** или **Зависимости**, после чего выбрав **Добавить ссылку** или **Добавить ссылку на проект**.

7. Выберите проект, содержащий код, который будет тестироваться, и нажмите **ОК**.



8. Добавьте код в метод модульных тестов.

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using System.IO;

namespace MyTestMod
{
    [TestClass]
    Ссылка: 0
    public class UnitTest1
    {
        public const string Expected = "Hello World!";

        [TestMethod]
        Ссылка: 0
        public void TestMethod1()
        {
            using (var sw = new StringWriter())
            {
                Console.SetOut(sw);
                MyTest.program.Main();

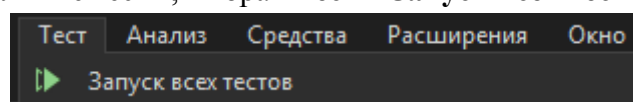
                var result = sw.ToString().Trim();
                Assert.AreEqual(Expected, result);
            }
        }
    }
}



```

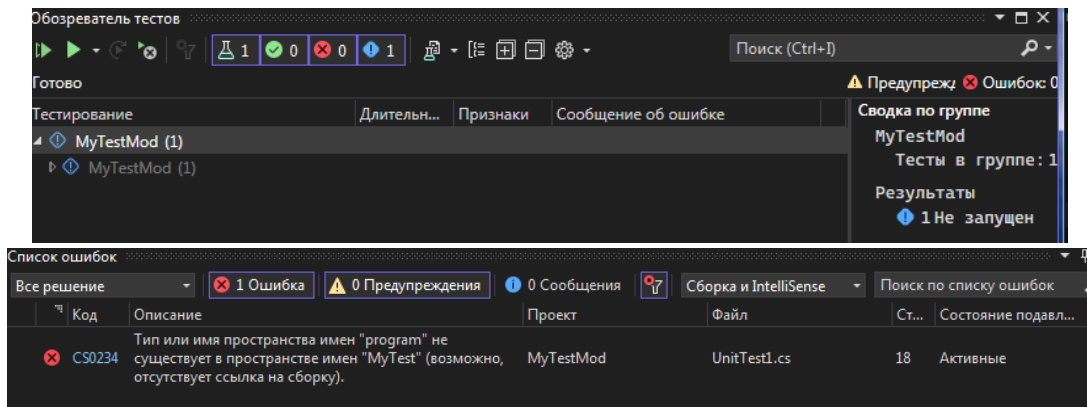
Задание 2 - Запуск модульных тестов

Технология выполнения работы:

1. Запустите модульные тесты, выбрав **Тест**→**Запуск всех тестов**



Иконка  показывает, что тест не запускался. Красный значок  указывает на сбой теста. В нижней части обозревателя теста должны выводиться ошибки в проекте.




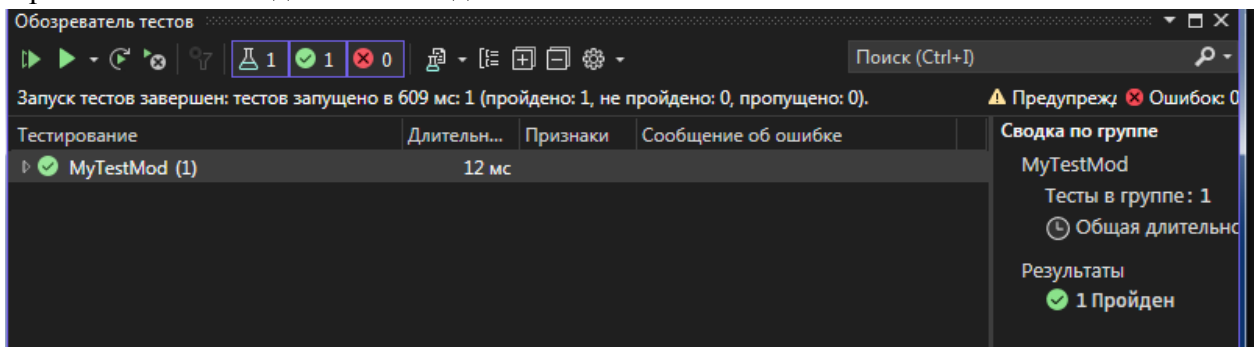
2. Исправьте ошибки и снова запустите тест.

```

1  using Microsoft.VisualStudio.TestTools.UnitTesting;
2  using System;
3  using System.IO;
4
5  namespace MyTestMod
6  {
7      [TestClass]
8      public class UnitTest1
9      {
10         public const string Expected = "Hello World!";
11
12         [TestMethod]
13         public void TestMethod1()
14         {
15             using (var sw = new StringWriter())
16             {
17                 Console.SetOut(sw);
18                 MyTest.Program.Main();
19
20                 var result = sw.ToString().Trim();
21                 Assert.AreEqual(Expected, result);
22             }
23         }
24     }
25 }
26

```

3. После завершения зеленый флажок  указывает, что тест пройден. В нижней части обозревателя теста не должно выводиться никаких ошибок



4. После успешного завершения теста посмотрите время его исполнения (в нашем случае это 12 мс).

Контрольные вопросы

1. Каковы цель и основные задачи модульного тестирования?
2. Назовите две основные проблемы, возникающие при модульном тестировании.
3. Дайте понятие модуля и его границ.
4. Характеристика процесса тестирования классов как модулей.
5. Определение степени полноты тестирования класса.
6. Протоколирование состояний объектов и их изменений.
7. Тестирование изменений, внесённых в исходный код.
8. Подходы к проектированию тестового окружения.

ЛАБОРАТОРНАЯ РАБОТА 11 (2 часа)

Тема: Выполнение функционального тестирования

Цель: изучение вопросов, связанных с функциональным тестированием. 13.2

Технология выполнения работы:

Задание 1 – Реализуйте функциональное тестирование способом разбиения по эквивалентности

Технология выполнения:

1. Определите входные данные
2. Определите выходные данные
3. Сформулируйте классы эквивалентности исходных данных алгоритма задачи.

Результат занести в таблицу

Входное условие	Правильные классы эквивалентности	Неправильные классы эквивалентности

4. Разработайте тестовый вариант для каждого класса эквивалентности

Варианты задания 1

1. Определить размер аванса, выплачиваемого каждому работнику подразделения с табельными номерами 11 по 51.
2. Протестировать поле даты в форме.
3. Программа проверяет пароль пользователя. Он должен быть 8 символов и содержать заглавные и прописные латинские буквы, цифры.
4. В приложении Microsoft Paint есть опция «Изменить размер» — «Наклон», которая принимает значения -89... 89. Составьте классы эквивалентности.
5. Протестировать поля логин и пароль при регистрации пользователя, если граница 8.
6. Протестировать поля логин и пароль при авторизации пользователя, если логин – test, пароль – 1234.
7. Протестировать поле для ввода денежных средств.
8. Протестировать отправку письма в смартфоне.
9. Определить период отпусков работников подразделения с табельными номерами 25..100.
10. Протестировать текстовое поле в форме.

Задание 2 – Реализуйте функциональное тестирование способом анализа граничных значений

Технология выполнения:

Реализуйте анализ граничных значений исходных данных поставленной задачи:

- определите минимальное граничное значение;
- определите максимальное граничное значение;
- определите допустимые входные данные;
- определите недопустимые входные данные;
- составьте тестовые варианты.

Варианты задания 2

1. Система просит пользователя ввести возраст. В зависимости от того, является ли пользователь совершеннолетним или нет, отображается различный контент.
2. Протестировать текстовое поле «Имя», которое принимает длину от 6 до 12 символов.
3. Протестировать поле «Электронная почта» в форме «Гостевая книга» веб-сайта.

4. Онлайн-продажа билетов в кинотеатр на фильмы с рейтингом «18+».
5. Даны числа a, b, c, d. Требуется упорядочить их по возрастанию.
6. Даны числа a, b, c, d. Требуется упорядочить их по убыванию.
7. Протабулировать функцию на интервале [-10; 15] с шагом 0,2.
8. Проверить дату как дату формата дд.мм.гггг.
9. Проверить услуги страховой компании: клиент может застраховать жизнь при возрасте 18 до 60 лет.
10. При нажатии на кнопку «Загрузить», система должна проверять размер загружаемого файла до 100 Мб и в соответствии с этим отправлять на определенный ftp-сервер.

Задание 3 – Реализуйте функциональное тестирование способом таблицы решений

Технология выполнения:

1. Сформулируйте классы эквивалентности исходных данных алгоритма задачи
2. Разработайте тестовый вариант для каждого класса эквивалентности
3. Составьте программу и протестируйте ее
4. Выработайте рекомендации для корректировки тестируемой программы

Варианты задания 3

1. Найдите произведение нечетных чисел от -18 до 18.
2. Вычислите функцию $y = \ln(x+1)$, для каждого x из [0, 10] изменяющегося с шагом 1.
3. Вычислите функцию $y = \sqrt{x+2}$ для каждого x из [-2, 14] изменяющегося с шагом 1.
4. Вычислить $Y = 1 + 1/x + 1/x^2$. Область: $x^2 + y^2 = 1$ для $x \in [0;1]$; $y \in [0;1]$.
5. Найти сумму целых чисел из [20, 100], кратных 5.
6. Найти произведение чисел из [20, 50], некратных 3.
7. Вычислить $y = \frac{2x}{4-x^2}$. Область определения: $x \in \mathbb{R}, x \neq 2, x \neq -2$.
8. Вычислите функцию $Y = \frac{1-X}{\sqrt{X^2-105}}$, если $x^2 < 105$ выражение не имеет действительных решений
9. Найдите произведение чисел от -25 до 30, кратных 2. Исключить 0.
10. Вычислить $Y = \frac{\sin x}{x^2 - 9}$. Область определения: $x \in \mathbb{R}, x \neq 3, x \neq -3$.

Типовой пример: Вычислите $y = x \cdot \cos(x) + \sin(3x)$. Область: $y^2 + x^2 = 1$ для $x, y \in [-1;1]$.

Технология выполнения:

1. Выделили классы эквивалентности
2. Определили граничные значения этих классов
3. Выполнили тесты по проверке значения до границы, на границе и сразу после границы.
4. Вычислили количество тестов для проверки граничных значений: [количество границ] * 3 = 2*3=6

Входные условия	Правильные классы эквивалентности	Неправильные классы эквивалентности
Переменная X	-1...1	$x < -1$ $x > 1$
Тип переменной X - вещественный	2.9E-39..1.7E+38	Буква, символ, запятая в числе

5. Составили тестовые варианты (табл. 1)
6. Написали программу на языке C++.

```

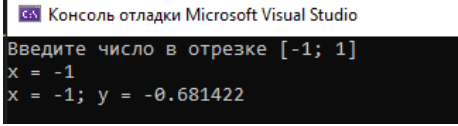
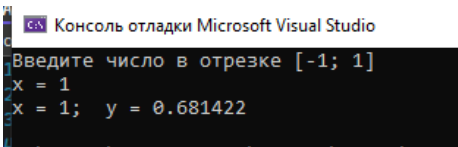
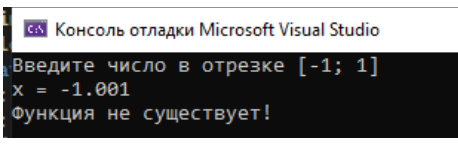
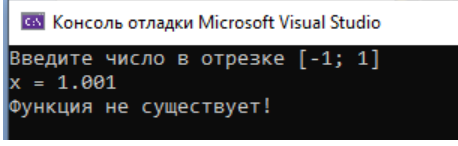
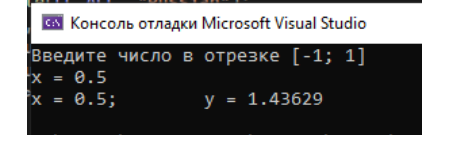
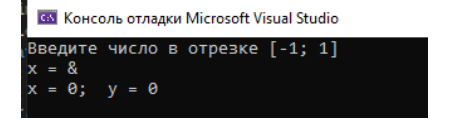
#include <iostream>
#include <math.h>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    float x, y;
    std::cout << "Введите число в отрезке [-1; 1]" << endl;
    std::cin >> x;
    if ((x >= -1) && (x <= 1)) {
        y = x * cos(x) + sin(3 * x);
        std::cout << "x = " << x << "; \t";
        std::cout << "y = " << y << endl;
    }
    else {
        std::cout << "Функция не существует!" << endl;
    }

    return 0;
}

```

7. Протестировали программу

Таблица 1 - Название теста - Вычисление функции

№ теста	X	Ожидаемый результат Y	Фактический результат	Результат тестирования	Предложения по исправлению найденных ошибок
1	-1	-0,7		успешный	-
2	1	0,7		успешный	-
3	-1,001	Функция не определена		успешный	-
4	1,001	Функция не определена		успешный	-
5	0,5	1,4		успешный	-
6	&	Ошибка ввода		неудачное	требуется ввести условие вычисления функции в случае символического ввода для числовой переменной x

Контрольные вопросы

1. Что понимается под функциональным тестированием?
2. В чём сущность тестирования «черного ящика» (blackbox)?
3. В чём сущность тестирования «белого ящика» (whitebox)?
4. В чём сущность тестирования «серого ящика» (gray-box)?
5. Методы отбора тестов для black-box тестирования.
6. Тестирование сценариев использования – юз-кейсов (usecases).
7. Тестирование классов эквивалентности.
8. В чём сущность попарного тестирования?

ЛАБОРАТОРНАЯ РАБОТА № 12 (2 часа)

Тема: Тестирование интеграции

Цель: изучение вопросов, связанных с интеграционным тестированием.

Задание - Реализовать интеграционное тестирование любого Интернет-магазина

Постановка задачи: требуется проверить, что совместимость web-сайта со сторонними сервисами:

- работу сторонних модулей: оплата, шаринг (совместное, коллективное использование и потребление различных вещей и услуг), карты.
- рекламу (просмотр, переходы по рекламе, аналитика).
- метрики (переходы по страницам, показы элементов, клики).

Технология выполнения:

1. Разработайте тест-план:
 - описание приложения;
 - список функций и описание тестируемой системы и её компонент;
 - объекты тестирования.
2. Составьте тест-кейс
3. Протестируйте web-сайт

Тест-кейс «Название сайта»

Номер теста	Цель теста	Описание теста	Ожидаемый результат
1			

4. Сделайте выводы и рекомендации.

Контрольные вопросы

1. Что понимается под интеграционным тестированием?
2. Дайте характеристику подходам к интеграционному тестированию: снизу-вверх; сверху вниз; большой взрыв.
3. В чём заключается принцип действия системы непрерывной интеграции?
4. В чём заключается интеграционное тестирование ASP.NET Core?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3 (2 часа)

Тема: Документирование результатов тестирования

Цель: изучить правила и последовательность составления итогового отчета о результатах тестирования web-приложения.

Задание - Провести документирование результатов ручного тестирования web-приложения.

Технология выполнения работы

1. Запустить ранее созданное или протестированное приложение.
2. Составить **итоговый отчет по результатам тестирования приложения:**
 - Указать общую информацию о тестируемом продукте:
 - название проекта;
 - модули, которые подверглись тестированию;
 - количество обнаруженных дефектов;
 - Указать, кто и когда тестировал программный продукт.
 - Описать тестовое окружение: ссылку на проект, браузер, операционную систему и другую информацию, конкретизирующую особенности конфигурации.
 - Указать общую оценку качества протестированного приложения и ее обосновать. Уровни качества: высокое (High), среднее (Medium), низкое (Low).
 - Определить показатель успешного прохождения тест-кейсов с помощью метрики по формуле

$$T^{SP} = \frac{T^{Success}}{T^{Total}} \cdot 100 \%,$$

- Графически (в виде круговой диаграммы) отразить процентное соотношение дефектов GUI и функциональных дефектов:

1) Реализовать **GUI – тестирование пользовательского интерфейса**

№	Тест-кейсы	Результаты теста
1	Проверить все элементы графического интерфейса на размер, положение, ширину, длину и прием символов или цифр.	успешно/дефект
2	Проверить, можете ли вы выполнить намеченную функциональность приложения, используя графический интерфейс	
3	Проверить сообщения об ошибках отображаются правильно	
4	Проверить на четкое разграничение различных разделов на экране	
5	Проверить, что используемый в приложении шрифт читабелен	
6	Проверить правильность выравнивания текста	
7	Цвет шрифта и предупреждающих сообщений эстетичны	
8	Убедиться, что изображения имеют хорошую четкость	
9	Убедиться, что изображения правильно выровнены	
10	Проверить расположение элементов графического интерфейса для разных разрешений экрана.	

- Вычислить процент дефектов GUI по формуле:

$$\text{Процент дефектов} = \frac{\text{Количества дефектов}}{\text{количество тестов}} * 100 \%$$

- 2) Реализовать **Functional** - тестирование, основанное на анализе спецификации функциональности

Функциональное тестирование интернет-магазина

При функциональном тестировании проверяется реализация функциональных требований, т.е. возможность программного продукта выполнять те функции, которые были

описаны в спецификациях на разработку. Каждый блок в отдельности необходимо тестировать на корректность реализации присущих ему функций.

№	Тест-кейсы	Результаты теста
1	Проверка баннера	успешно/дефект
2	Поиск товара	
3	

- Вычислить процент функциональных дефектов
 - Графически (в виде столбчатой диаграммы) отразить распределение дефектов по модулям.
 - Произвести детальный анализ качества всех модулей протестированного приложения с аргументацией выставленных уровней качества.

Модули	Уровень качества	Комментарии
Главная		
Каталог товаров		
Корзина		

- Привести список 3-5 наиболее критичных дефектов.
- Сформулировать рекомендации по улучшению качества программного продукта.

Контрольные вопросы:

1. Какая структура итогового отчета о результатах тестирования?
2. Что содержится в разделе Общая информация?
3. Что содержится в разделе Тестовое окружение?
4. Что содержится в разделе Рекомендации QA?
5. Что содержится в разделе Детализированная информация?
6. Что содержится в окончании отчета?

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Заботина, Н. Н. Проектирование информационных систем: учебное пособие / Н. Н. Заботина. — Москва: ИНФРА-М, 2020. — 331 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-004509-2. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1036508> (дата обращения: 10.02.2022). – Режим доступа: по подписке.

2. Рудаков, А. В. Технология разработки программных продуктов: учебник для студентов учреждений среднего профессионального образования, обучающихся по специальности "Программное обеспечение вычислительной техники и автоматизированных систем": [профессиональный модуль ПМ.03 "Участие в интеграции программных модулей" (МДК.03.01)] / А. В. Рудаков; А. В. Рудаков. – 12-е изд., стер. – Москва: Академия, 2018. – 208 с. – ISBN 9785446877928. – URL: <https://www.academia-moscow.ru/catalogue/4831/401005/>. – Текст: электронный.

3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учеб.пособие / Г.Н. Федорова. — М.: КУРС: ИНФРА-М, 2019. — 336 с. (Среднее Профессиональное Образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/989682> (дата обращения: 12.02.2022). – Режим доступа: по подписке.

Дополнительные источники:

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2021. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1189951> (дата обращения: 20.04.2022). – Режим доступа: по подписке.

Интернет-ресурсы:

1. Жизненный цикл программного продукта: [сайт]. – URL: <https://qaevolution.ru/zhiznennyj-cikl-programmnogo-obespecheniya> (дата обращения 10.05.2021). - Текст : электронный.

2. Моделирование на UML [сайт]. – URL: <http://book.uml3.ru/content> (дата обращения 10.03.2022). - Текст : электронный.

3. НОУ «ИНТУИТ» [сайт]. – Москва, 2003-2020.- URL: <http://www.intuit.ru/studies/courses/617/473/lecture/20968> (дата обращения 02.03.2022). -Текст : электронный.

Составитель
Белугина Светлана Викторовна

**МДК 02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

Методические указания по выполнению лабораторно-практических работ
для студентов очной формы обучения
по направлению специальности
09.02.07 «Информационные системы и программирование»

Публикуется в авторской редакции